

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

«На правах рукопису»

УДК 004.4

«До захисту допущено»

Завідувач кафедри

“ ____ ” _____ 20 ____

р.

Магістерська дисертація

зі спеціальності (спеціалізації) 121 «Інженерія програмного забезпечення»
(код і назва спеціальності)

на тему: Система координації наземних рухомих об'єктів за допомогою GPS

Виконав: студент 6 курсу, групи ІТ-з 83 мп

Громович Владислав Костянтинович

Науковий керівник: доктор військових наук, професор Шемаєв В.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

Консультант _____

(назва розділу)

(науковий ступінь, вчене звання, , прізвище, ініціали)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою
 Спеціальність (спеціалізація) – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Громовичу Владиславу Костянтиновичу

1. Тема дисертації «Система координації наземних рухомих об'єктів за допомогою GPS»

Науковий керівник дисертації: Шемаєв В.М. доктор військових наук, проф.,
 затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Строк подання студентом дисертації _____

3. Вихідні дані до проекту: Спроекувати та розробити систему для координації наземних рухомих об'єктів за допомогою GPS

4. Перелік завдань, які потрібно розробити : Аналіз актуальності проблеми та існуючих рішень, формування вимог до системи, розробка сценаріїв використання, вибір та обґрунтування елементів та технологій для реалізації системи та архітектури, розробка стартап-проекту

5. Орієнтовний перелік ілюстративного (графічного) матеріалу : Діаграма використання, ER-діаграма, схема бази даних, структурна схема, діаграма діяльності, діаграма послідовності, діаграма компонентів, діаграма дерево функцій

6. Орієнтовний перелік публікацій

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	завдання прийняв

8. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>08.09.2019</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>20.09.2019</i>	
3.	<i>Постановка задачі</i>	<i>29.09.2019</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>01.10.2019</i>	
5.	<i>Алгоритмізація задачі</i>	<i>12.10.2019</i>	
6.	<i>Розробка програмного забезпечення</i>	<i>24.10.2019</i>	
7.	<i>Налагодження програми</i>	<i>06.11.2019</i>	
8.	<i>Виконання графічних документів</i>	<i>19.11.2019</i>	
9.	<i>Оформлення пояснювальної записки</i>	<i>28.11.2019</i>	
10.	<i>Подання ДП на попередній захист</i>	<i>03.12.2019</i>	
11.	<i>Подання ДП на основний захист</i>	<i>17.12.2019</i>	

Студент _____ Громович В.К.

Науковий керівник проекту _____ Шемаев В.М.

РЕФЕРАТ

Громович В.К., магістерська дисертація на тему: «Система координації наземних рухомих об'єктів за допомогою GPS». НТУУ «КПІ» ім. Сікорського, Київ, 2019.

Проект складається з семи розділів, містить 105 сторінок, 34 рисунків, 40 таблиць, 8 аркушів графічного матеріалу формату А3, 24 джерел.

Об'єктом дослідження є система для координації наземних рухомих об'єктів.

Мета дисертації - розробка системи для полегшення процесу координації наземних рухомих об'єктів та створення суспільно-корисного продукту з більшим функціоналом та ліпшими характеристиками.

У магістерській дисертації розроблено систему, де будь який користувач може визначити свою геопозицію та провести координацію своїх дій з іншими користувачами. У ході створення даної системи було ретельно проведено огляд наявних аналогів, побудована структура бази даних, обрані та обґрунтовані елементи та технології для реалізації, визначені сценарії використання системи та сформовані вимоги до неї, описано реалізацію архітектури та проведено дослідження стартап потенціалу системи.

Наукова новизна дисертації полягає у об'єднанні навігаційних, координаційних та програмних технологій з метою покращення управління об'єктами. Практичне значення полягає у розробленні системи для координації об'єктів, що дозволяє координувати великі по кількості групи об'єктів в режимі реального часу на карті.

Розроблена система є багатокористувацькою, конкурентоспроможною, має широкий функціонал, зручний інтерфейс та легка для модифікації задля спеціальних потреб користувача.

Ключові слова: GPS, координація, місцезнаходження, система координації, геопозиція, GPS-трекер.

ABSTRACT

Gromovich VK, Master's Thesis on "System for coordinating terrestrial moving objects with GPS ". NTUU " Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, 2019.

The project consists of seven sections, containing 105 pages, 34 figures, 40 tables, 8 sheets of A3 graphic material, 24 sources.

The object of study is a system for coordinating terrestrial moving objects.

The purpose of the dissertation is to develop a system to facilitate the coordination of terrestrial moving objects and to create a public utility product with greater functionality and better performance.

The dissertation is devoted to the development of a system where any user can determine their location and coordinate with other users. During the creation of this system, a thorough review of the available analogues was conducted, the structure of the database was built, the elements and technologies for implementation were selected and justified, the scenarios of the system usage and requirements were defined, the implementation of the architecture was described and the startup potential of the system was conducted.

The scientific novelty of the dissertation is to combine navigation, coordination and software technologies to improve object coordination. The practical value is to develop a system for coordinating objects that allows you to coordinate large numbers of groups of objects in real time on the map.

The developed system is multi-user, competitive, has wide functionality, user-friendly interface and is easily modified for special needs of the user.

Keywords: GPS, coordination, location, coordination system, geopositioning, GPS tracker.

ЗМІСТ

ВСТУП	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	6
1.1 Опис предметного середовища	6
1.2 Аналіз існуючих рішень	8
1.3 Висновок до розділу	23
2 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	24
2.1 Функціональні та не функціональні вимоги	24
2.2 Бізнес правила	30
2.3 Цілі та задачі розробки	35
2.4 Опис процесу діяльності	36
2.5 Структурна схема	38
2.6 Висновок до розділу	39
3 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ	40
3.1 Основний сценарій використання системи	40
3.2 Сценарій авторизації користувача	40
3.3 Сценарій реєстрації	41
3.4 Сценарій входу на сторінку за картою	42
3.5 Сценарій визначення місцезнаходження	42
3.6 Сценарій побудови маршрутів	43
3.7 Сценарій створення групи користувачів	43

3.8 Сценарій обміну даними	44
3.9 Сценарій додавання та видалення користувачів	45
3.10 Сценарій видалення повідомлень	45
3.11 Висновок до розділу	46
4 ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ	47
4.1 Мови програмування	47
4.1.1 JavaScript	47
4.1.2 CSS	49
4.1.3 HTML	51
4.1.4 Java	52
4.2 База даних	59
4.3 Фреймворки та бібліотеки	60
4.3.1 Hibernate	60
4.3.2 Google Maps API	62
4.3.3 Vue.js	63
4.3.4 Spring Data JPA	65
4.3.5 Spring Security	65
4.4 Висновок до розділу	66
5 СТРУКТУРА БАЗИ ДАНИХ	67
5.1 ER- діаграма	67
5.2 Схема бази даних	67

5.3 Висновок до розділу	69
6 АРХІТЕКТУРА ТА ЛОГІКА СИСТЕМИ	70
6.1 Розробка архітектури системи	70
6.1.1 REST	70
6.1.2 AJAX	73
6.2 Діаграма послідовності	75
6.2.1 Процес авторизації	76
6.2.2 Процес реєстрації	77
6.2.3 Процес визначення місцезнаходження	78
6.2.4 Процес створення групи	78
6.3 Шифрування	80
6.4 Висновок до розділу	81
7 СТАРТАП-ПРОЕКТ	82
7.1 Опис ідеї проекту	82
7.2 Технологічний аудит ідеї проекту	84
7.3 Аналіз ринкових можливостей запуску стартап-проекту	85
7.4 Розроблення ринкової стратегії проекту	95
7.5 Розроблення маркетингової програми стартап-проекту	98
7.6 Висновок до розділу	102
ВИСНОВКИ	103
ПЕРЕЛІК ПОСИЛАНЬ	104

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GPS – (англ. Global Positioning System) система глобального позиціонування

WHOIS — протокол запит/відповідь, що базується на TCP

IP-адреса — (англ. Internet Protocol address) — це ідентифікатор (унікальний числовий номер) мережевого рівня

MAC-адреса — (англ. Media Access Control) — це унікальний ідентифікатор, що зіставляється з різними типами устаткування для комп'ютерних мереж

UTC — (англ. Coordinated Universal Time) стандарт, за яким суспільство регулює годинники й час

АСУ — автоматизована система управління

GPRS — (англ. General Packet Radio Service) загальний сервіс пакетної радіопередачі

СУБД — (англ. Database Management System, DBMS) система управління базами даних

CSS — (англ. Cascading Style Sheets) каскадні таблиці стилів

HTML— (англ. HyperText Markup Language) мова гіпертекстової розмітки

JVM — (англ. Java Virtual Machine) віртуальна машина Java

ER-діаграма — (англ. Entity-relationship model або entity-relationship diagram) модель сутність-зв'язок

Ajax — (англ. Asynchronous Javascript and XML) асинхронний JavaScript і XML

REST — (англ. Representational State Transfer) передача стану уявлення

ВСТУП

Система координації наземних рухомих об'єктів за допомогою GPS – це веб-ресурс, за допомогою якого користувачі можуть здійснювати координацію наземних рухомих об'єктів. Ця система призначена для прийому і передачі даних, які застосовуються для моніторингу розташування об'єкта. Вона використовує систему геолокації GPS для визначення місцезнаходження того чи іншого об'єкта.

Актуальність цієї теми полягає в тому, що необхідність відстежувати місцеположення як своє, так і інших користувачів виникає у великої кількості людей. Але, якщо для відстежування своєї позиції існує багато як застосувань і різних онлайн карт, то для відстежування декількох об'єктів водночас і можливості координувати свої дії одне з одним – таких вирішень є небагато, і не всі задовольняють різні потреби користувачів. Існуючі роботи не забезпечують зручної можливості координувати дії об'єктів в середині цих робіт, і користувачам потрібно знаходити інші шляхи для вирішення цього питання, а саме: комбінувати різні програми або можливості для координації. Також, існують компанії, які потребують відстеження та координацію своїх вантажних автомобілів.

Тому розробка веб-ресурсу, який міг би працювати на всіх платформах та вирішував всі проблеми попередніх робіт є актуальною.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Предметним середовищем є геолокація. Геолокація - це процес пошуку, визначення та надання точного розташування комп'ютера, мережевого пристрою чи обладнання. Це дозволяє розташувати пристрій на основі географічних координат та вимірювань. Геолокація зазвичай використовує Глобальну систему позиціонування (GPS) та інші пов'язані з ними технології для оцінки та визначення географічного положення. [1]. Геолокація працює за допомогою попередньо вбудованого GPS у пристрої, який поширює поздовжні та широтні координати пристрою. Координати ідентифікуються на карті, щоб забезпечити повну адресу, яка зазвичай включає країну, місто, місто / колонію, назву будівлі та адресу вулиці. Крім GPS, геолокація також може бути ідентифікована через адресу Інтернет-протоколу (IP), адресу управління доступом до засобів масової інформації (MAC), радіочастотні системи (RF), дані формату файлів зображення (EXIF) та інші бездротові системи позиціонування. (рис.1.1)[2].

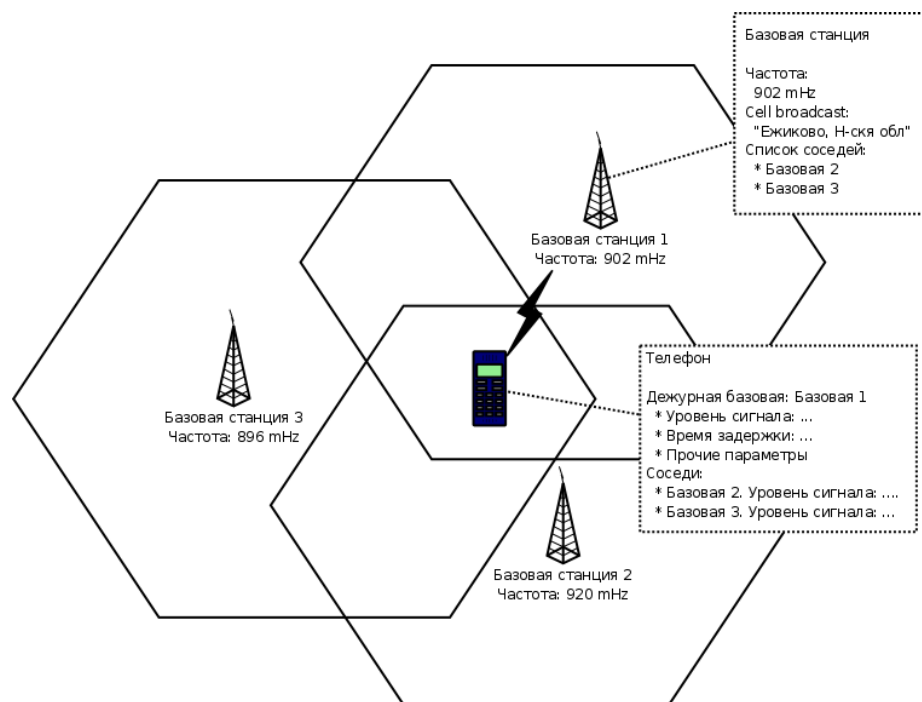


Рисунок 1.1— Триангуляція

Точність цих методів набагато менша, ніж GPS, але за останні роки значно зросла. Ці технології значно відрізняються від попередніх радіолокаційних методів, таких як радіопеленгація.

Геолокацію комп'ютерів та інших пристроїв Інтернету можна здійснити шляхом підключення до конкретної реальної IP-адреси, MAC-адреси або номера RFID, що зберігаються в пам'яті пристрою, за допомогою ідентифікаторів програмного забезпечення або сучасних методів, стеганографії, позиціонування Wi-Fi, друку пристроїв, друку полотна, координат пристрою GPS, тощо. Геолокація в Інтернеті часто працює через сервер WHOIS і вимагає IP-адреси фізичної адреси власника.

Термін "геолокація" також відноситься до відстеження міграції тварин та спостереження за навколишнім середовищем пристроями, приєднаними до тіл тварин (рис. 1.2) [3], включаючи імплантовані транспондери ідентифікації та реєстратори даних.



Рисунок 1.2 – Приклад GPS-трекера для тварин

Інформація про IP-адресу геолокаційного програмного забезпечення може включати країну, регіон, місто, поштовий індекс, широту, довготу та часовий пояс. Іноді можна отримати більше даних з однієї IP-адреси, включаючи доменне ім'я, швидкість з'єднання, ім'я провайдера, мову, проксі-сервер, ім'я організації, робочу та домашню адресу та телефон.

1.2 Аналіз існуючих рішень

В ході аналізу існуючих рішень було виявлено не багато застосунків з необхідною функціональністю. Нижче буде приведено низка схожих.

1. Застосунок «ГДЕ Я»(рис.1.3)[4].

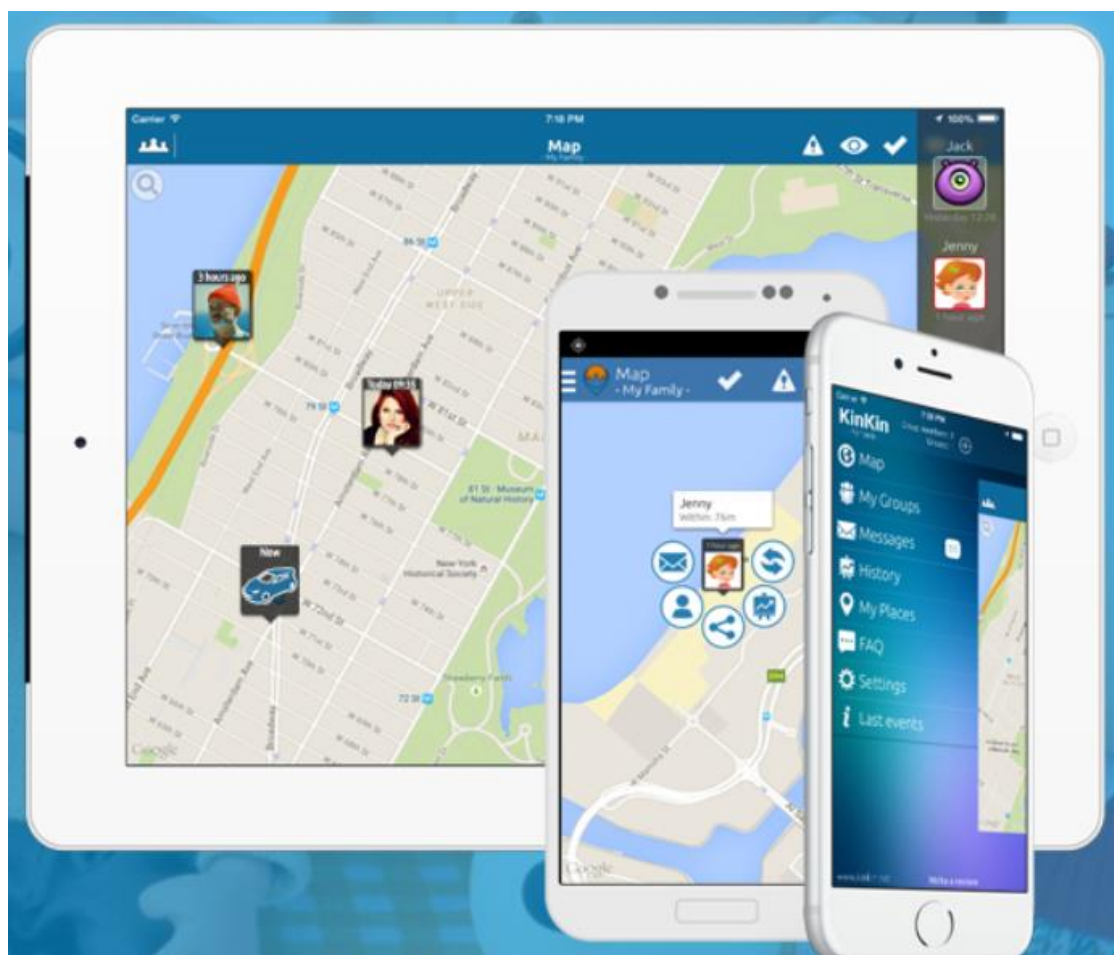


Рисунок 1.3 – Приклад відображення карти застосунку «Где Я» на різних системах

Застосунок «ГДЕ Я» є хорошим прикладом GPS-трекера, який дозволяє бачити всю вашу сім'ю на карті зі свого телефону, планшета чи безпосередньо на веб-сайті,

створювати власні місця на карті для себе чи всієї родини та обмінюватись повідомленням, показувати один раз для всієї родини або в особистому контакті історію руху члена сім'ї протягом певного періоду часу, додавати не лише членів сім'ї, але й своїх друзів і навіть автомобілі, щоб створити маршрут від вашого поточного місцезнаходження до членів сім'ї або будинку / роботи.

Користувачі встановлюють додаток на свої мобільні телефони та мобільні телефони всіх членів сім'ї. Телефони самі передають свої координати у фоновому режимі до місця, де вони зберігаються. Користувачі також можуть переглядати поточний стан члена сім'ї або історію за будь-який день. На даний момент підтримується iPhone / iPad / iPod та всі Android-пристрої.

Має низький рейтинг та купу негативних відгуків.

2. Застосунок «Enduro Tracker» (рис.1.4)[5].

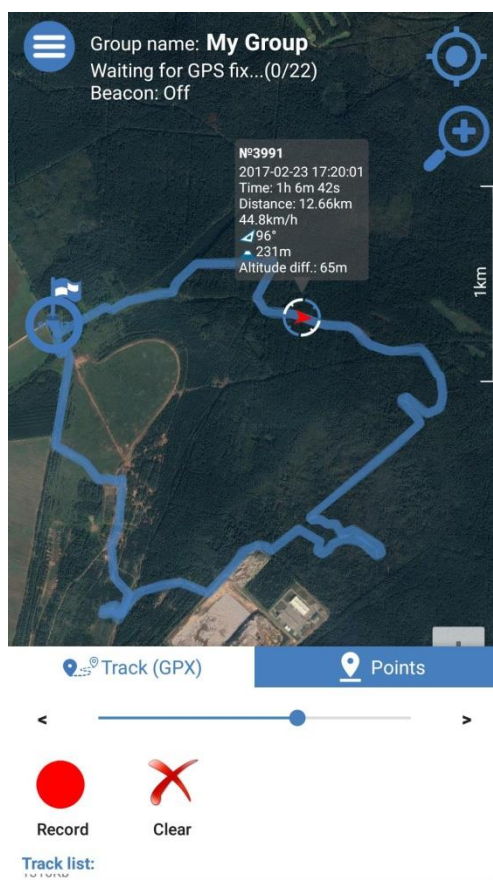


Рисунок 1.4 – Приклад відображення карти застосунку «Enduro Tracker»

Спочатку розроблений як GPS-трекер, надає можливість користувачам переглянути місця своїх друзів у подорожах ендуро. Його також можна легко використовувати для відстеження рухів автомобіля, дитини чи навіть вашої домашньої тварини, або для будь-якого іншого моніторингу положення членів групи в реальному часі.

3. Застосунок «GPS Лама» (рис.1.5)[6].

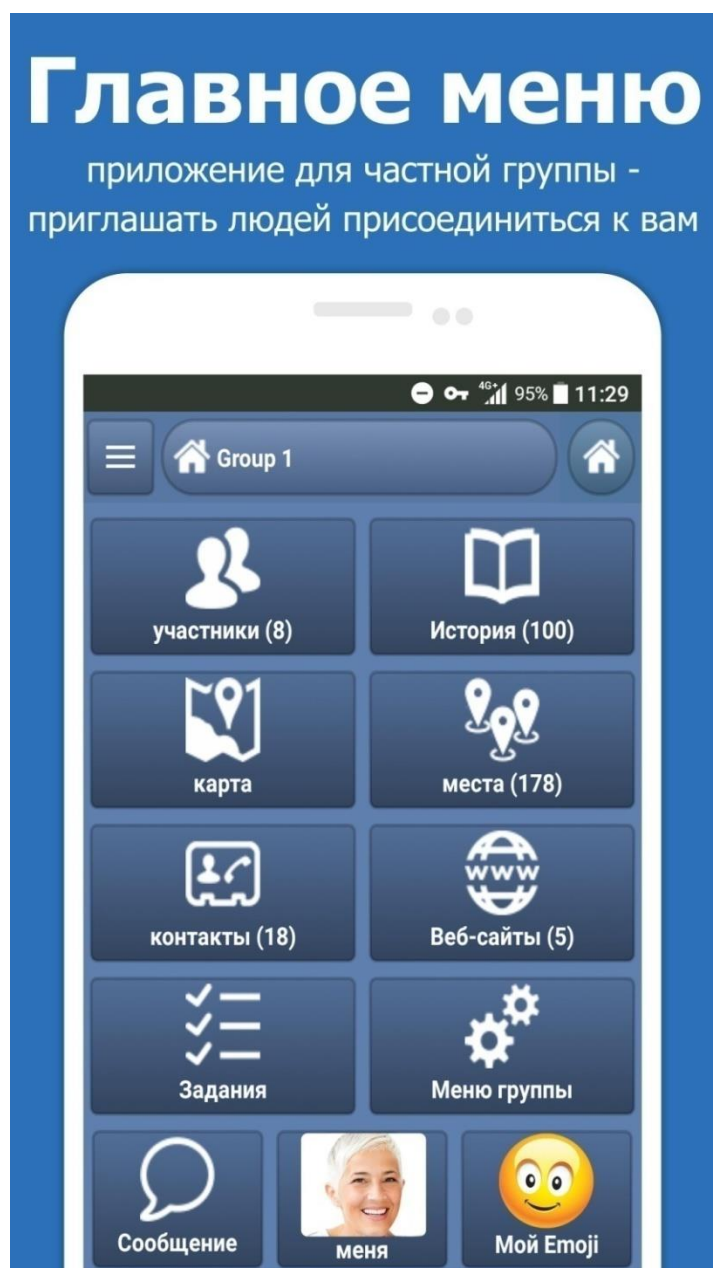


Рисунок 1.5 – Приклад головного екрану і функцій

Застосунок є технічно сучасний, приватний, захищений та був удостоєний престижної премії інновацій британським урядом.

Застосунок « GPS Лама» має такі можливості:

- застосунок продовжує працювати, коли телефон не працює (не діє);
- застосунок дозволяє відстежувати кілька телефонів у приватній групі, створювати окремі приватні групи для сім'ї, роботи;
- застосунок містить кнопку SOS(рис. 1.6)[6];
- застосунок містить пульс, який повідомляє користувачів, якщо телефон у їхній групі перестає працювати (немає сигналу або застосунок зупинено);

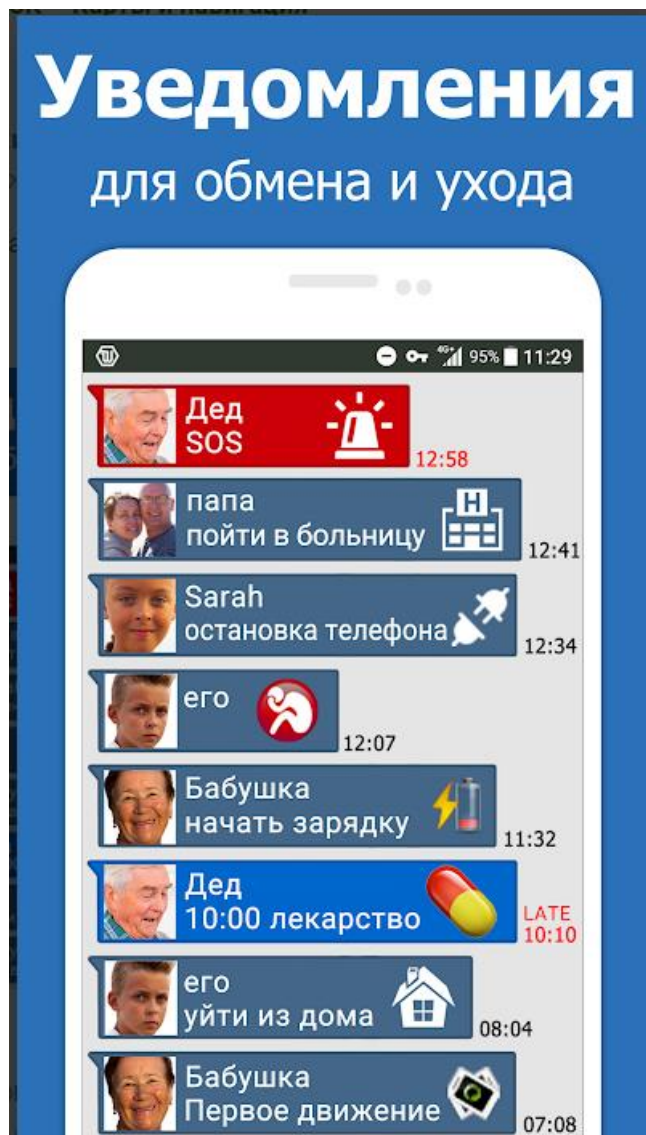


Рисунок 1.6 – Приклад оповіщення застосунку «GPS Лама»

- застосунок також відстежує заряд батареї іншого телефону і відправляє «попередження про низький рівень заряду»;
- якщо користувач турбується про те, щоб кудись поїхати, наприклад, агент з нерухомості відвідує ізольовану власність, вона може натиснути «Поділитися місцем розташування», щоб люди з групи могли бачити ваші GPS-треки;
- люди в групі отримують живі повідомлення: коли ви приходите на місце, якщо ваш телефон перестає працювати (вимкнений або пропав сигнал); і коли ти покинеш це місце;
- користувачі можуть організувати «місця», і вони будуть доступні для інших людей в групі(рис.1.7)[6];



Рисунок 1.7 – Приклад карти застосунка

- користувачі можуть отримувати інформацію про положення GPS кожні 60 секунд, навіть коли телефон спить;
- за стосунок має два датчики: датчик руху повідомляє про те, коли телефон останній раз виймали, а інший датчик показує заряд акумулятора;

- користувач може покласти телефон в машину або фургон та віддалено стежити за GPS-адресами, щоб отримувати автоматичні оповіщення в реальному часі, розуміти, коли автомобіль в'їжджає чи залишає місце, а також отримувати сповіщення про датчики руху та попередження швидкості;
- в застосунку є спеціальна кнопка «Перейти», що дозволяє людині вибрати пункт призначення (варіант передбачуваного часу прибуття);
- користувачі отримують повідомлення в прямому ефірі, коли людина починає свою поїздку і знову, коли закінчує свою поїздку;
- застосунок дозволяє управляти і координувати роботу кількох водіїв, кількох транспортних засобів і транспортних поїздок;
- застосунок дозволяє створювати і ділитися: телефонна книга; веб-закладки; карта місць: всякий раз, коли користувач міняє інформацію, додаток автоматично оновлює всіх інших членів в групі.

Існує також багато рішень, які були створені для координації військових об'єктів. Збройні сили стикаються з добре налаштованим противником, який використовує широкий спектр традиційних та асиметричних тактик у складних просторах. Ця проблема підкреслює нагальну необхідність швидкого вдосконалення вертикальної та горизонтальної інтеграції та розподілу повноважень військового командування як всередині армії, так і між військами у спільному просторі та між організаціями та країнами у взаємодіючих та багатонаціональних просторах. Більше не прийнято мати будь-яку військову силу, яка діє незалежно в одному географічному районі. Взаємодія - це здатність систем, підрозділів або сил надавати дані, інформацію, матеріальні компоненти та послуги та отримувати їх від інших систем, підрозділів чи сил та використовувати все це для ефективної спільної роботи.

Інформаційні системи управління боротьбою - це пристрої та інструменти, які збирають, обробляють, зберігають, відображають та поширюють інформацію. Сюди входять комп'ютери, апаратне, програмне забезпечення та комунікації, а також

методи та процедури їх використання. Ключовим елементом загальної стратегії армії систем управління бойовими діями є подолання епохи нових вертикальних можливостей та злиття багатофункціональних, основних систем зв'язку армії в одну унікальну.

Велика кількість країн вже розпочала модернізацію свого обладнання, а деякі найкращі товари були знайдені в ході пошуку доступних аналогів.

1. Навігаційний комплекс топогеодезичного і тимчасового забезпечення СН-3210 "Базальт-К"(рис. 1.8)[7].

Навігаційний комплекс СН-3210 забезпечує рішення таких задач:

- визначення експлуатаційних опертивних параметрів навігації об'єктів і часу в UTC;
- відображення навігаційних параметрів об'єктів на цифровій карті місцевості;
- збереження та всебічна обробка інформації про навігацію та карту;



Рисунок 1.8 – Комплекс СН-3210

- планування маневрів та дистанційне керування об'єктами маневрів;
- обмін оперативною навігаційною інформацією з взаємодіючими об'єктами;
- робота з іншими датчиками навігації на інтерфейсі RS-232 та можливість роботи з іншими компонентами системи;
- автоматичне управління функцією пристрою, відображення помилок;
- робота з векторною цифровою картою місцевості;
- введення, зберігання, вибір та відображення у вибраному масштабі карти потрібної області;
- масштабування та відображення карти за командою оператора;
- відображення, реєстрацію і контроль параметрів розташування та переміщення взаємодіючих об'єктів на карті вибраного масштабу;
- планування маршрутів з урахуванням оперативно-тактичних завдань в графічному і табличному виконанні;
- збереження, редагування і використання в подальшому пройдених маршрутів в якості запланованих;
- розрахунок параметрів навігації (напрямки руху / часу / швидкості) виходу в задану точку;
- вибір і нанесення маршруту на карту;
- розрахунок азимуту, відстані, часу подорожі до точки, тривалість перебування в точці, очікування, рекомендована швидкість;
- видачу повідомлень про прибуття в спеціальні точки маршруту;
- спільній роботі з аналоговими і цифровими радіостанціями для реалізації інформаційних автоматизованих систем;
- обчислення дирекційних кутів і відстаней до об'єктів і між об'єктами, які відображені на карті, а також рішення прямої і зворотної геодезичної задачі;
- рішення сервісних завдань (топогеодезичні завдання в просторі і спеціальні завдання проходження маршруту);
- перерахунок та відображення дорожнього кута в поділках кутоміра;
- підключення зовнішньої камери через послідовний порт USB, і відображення на екрані у відповідному формулярі зображення зовнішньої обстановки.

2. Автоматизована система управління військами «Андромеда-Д»[8].



Рисунок 1.9 – Система «Андромеда-Д»

За допомогою системи "Андромеда-Д" дані передаються на відстані приблизно 100 км через усі з'єднання - від керівника маневру до винищувача війська до бойової машини та окремого бійця.

Система "Андромеда-Д" пропонує стабільні канали для обміну інформацією та дозволяє швидше передавати інформацію в безпечних та кодованих режимах. "Андромеда-Д" забезпечує багатосмугові радіо виклики на різних діапазонах, закриту телефонію, систему зашифрованої передачу даних та роботу в режимі відеоконференції.

3. Автоматизована система управління військами «Альянс» (рис. 1.10)[9].

Автоматизована система управління сухопутними військами «Альянс» призначена для ефективного управління угрупованнями сухопутних військ.

В основу концепції АСУ «Альянс» покладено принцип, що кожна одиниця техніки або військовослужбовець - це інформаційний вузол в системі.

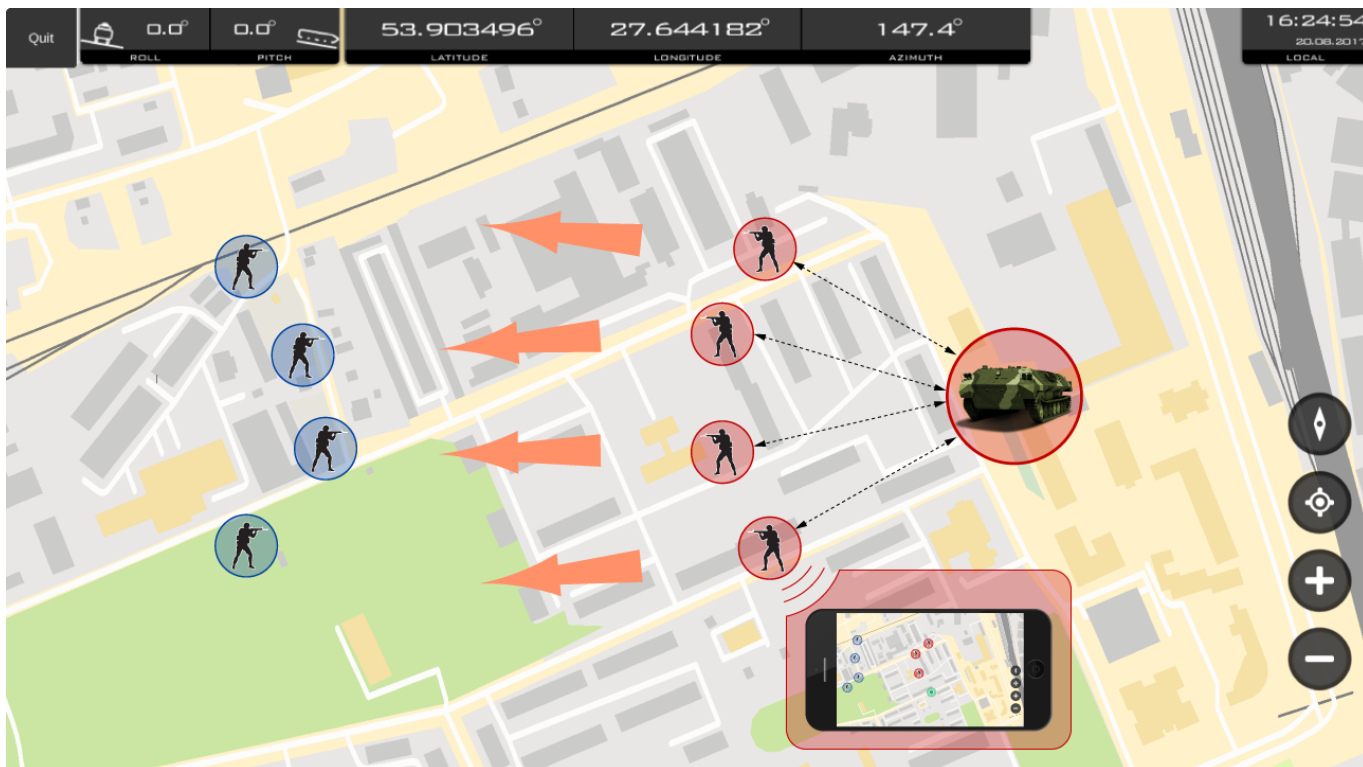


Рисунок 1.10 – Приклад карти системи «Альянс»

Як результат, кожен командувач і командир на своєму рівні управління знаходиться в постійній обізнаності про стан противника, дії підлеглих військ і здійснює ефективну їх координацію в режимі реального часу.

У зв'язку з тим, що АСУ «Альянс» побудована на модульній основі, обмін інформацією між модулями від нижніх до стратегічних рівнів забезпечується в режимі реального часу. При необхідності розширити функціональність системи для вирішення особливих завдань можна за рахунок інтеграції різних інструментів для розрахунку логічних і математичних задач, а також інструментів для роботи з векторними, растровими, оцифрованими і тривимірними картами місцевості, що забезпечують, в тому числі партійні постачальники.

Користувач може використовувати автоматизовану систему управління військами «Альянс» на будь-якому пристрої, будь то смартфон, планшет, ноутбук, настільний комп'ютер або розумний годинник. Неважливо, яка операційна

платформа там встановлена - Android, AndroidWear, iOS, Windows, Linux. Розробник не пов'язував Альянс з яким-небудь певним апаратним устаткуванням і програмним забезпеченням.

4. Комплекс планування, розвідки і координації MAP21(рис. 1.11)[9].

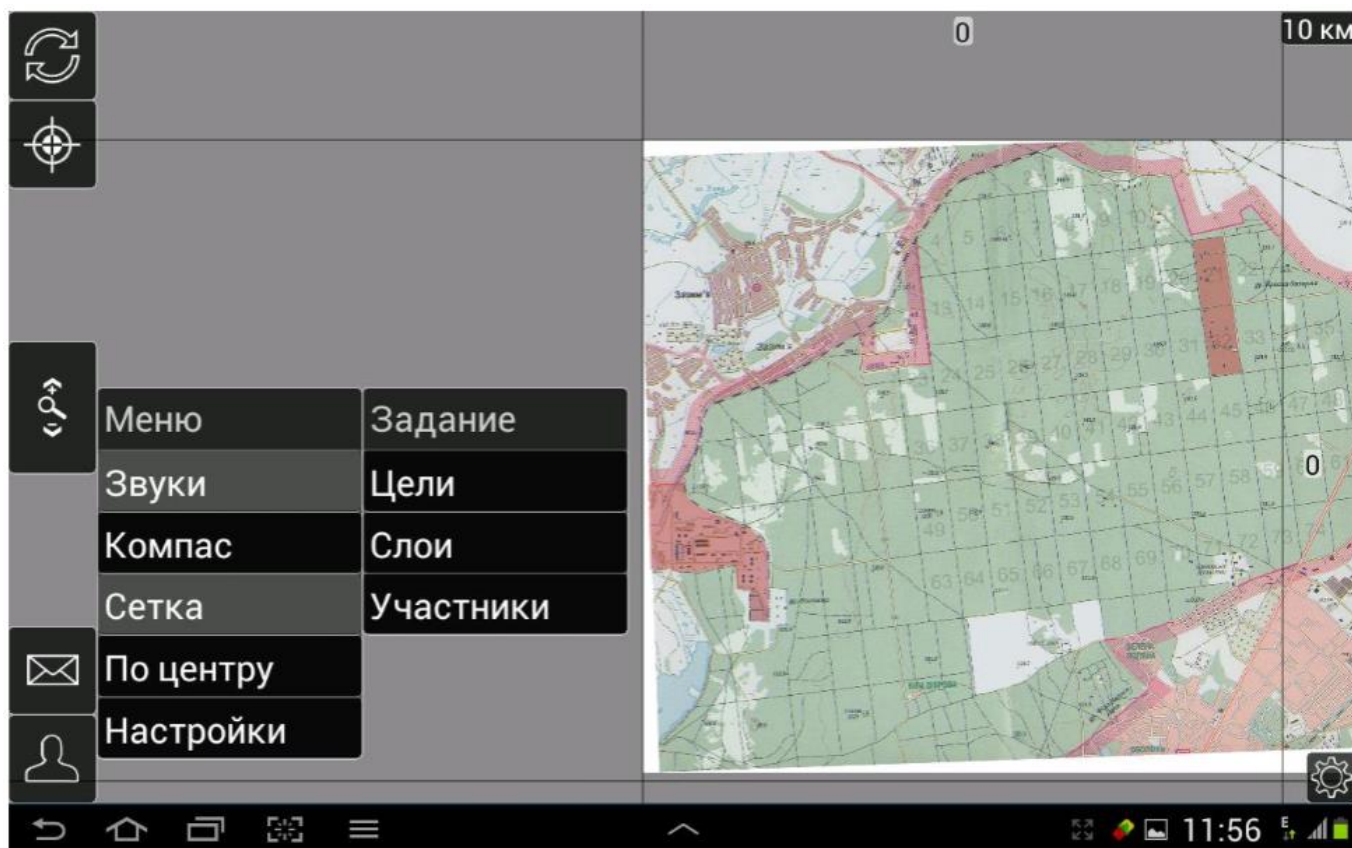


Рисунок 1.11 – Приклад відображення основного меню комплексом «МАР 21»

Програмний комплекс «МАР21» є потужним інструментом для попереднього планування заходів, застосування розвідданих і оперативного управління структурними підрозділами. Цільова аудиторія застосування - тактичні страйкбольні і пейнтбольні ігри, змагання з орієнтування і квести(рис.1.12)[9]. В останніх оновленнях в комплексі приділили увагу питанням безпеки та покращили питання шифрування і аутентифікації користувача, що разом з доданою автономною системою польової передачі даних надало можливість застосовувати комплекс «МАР 21» в реальних бойових діях та під час тренувань армії. Ці оновлення були виконані на замовлення силових структур.



Рисунок 1.12 – Приклад карти застосування «MAP 21»

Основною функціональністю комплексу є взаємне відображення користувачів на карті місцевості, використання системи GPS-позиціонування, автоматична синхронізація між користувачами та їхніх заданих об'єктів (таких як: мітки дружніх, ворожих, нейтральних і інших юнітів; зони на карті, намальовані за допомогою різних геометричних фігур; поточні завдання для кожного користувача; маршрути руху і інша інформація), а також постановка цілей і видача оперативних замовлень від одного користувача іншому, при цьому розмежуючи їх права на основі певної ієрархічної структури.

Унікальною особливістю комплексу є система шарів, тригерів і скриптованої мови, які дозволяють автоматизувати процес навчання на рівні програми. Комплекс дає можливість заздалегідь спланувати різні етапи заходи на різних шарах, кожен з яких можна приховати і відобразити окремо в потрібний момент, автоматично змінити інформацію на карті і задати таргетинг (рис.1.13) [9] користувачам,

грунтуючись на інформації про їхнє становище в просторі і взаємодії з об'єктами на карті за допомогою тригерів і гнучких сценаріїв.

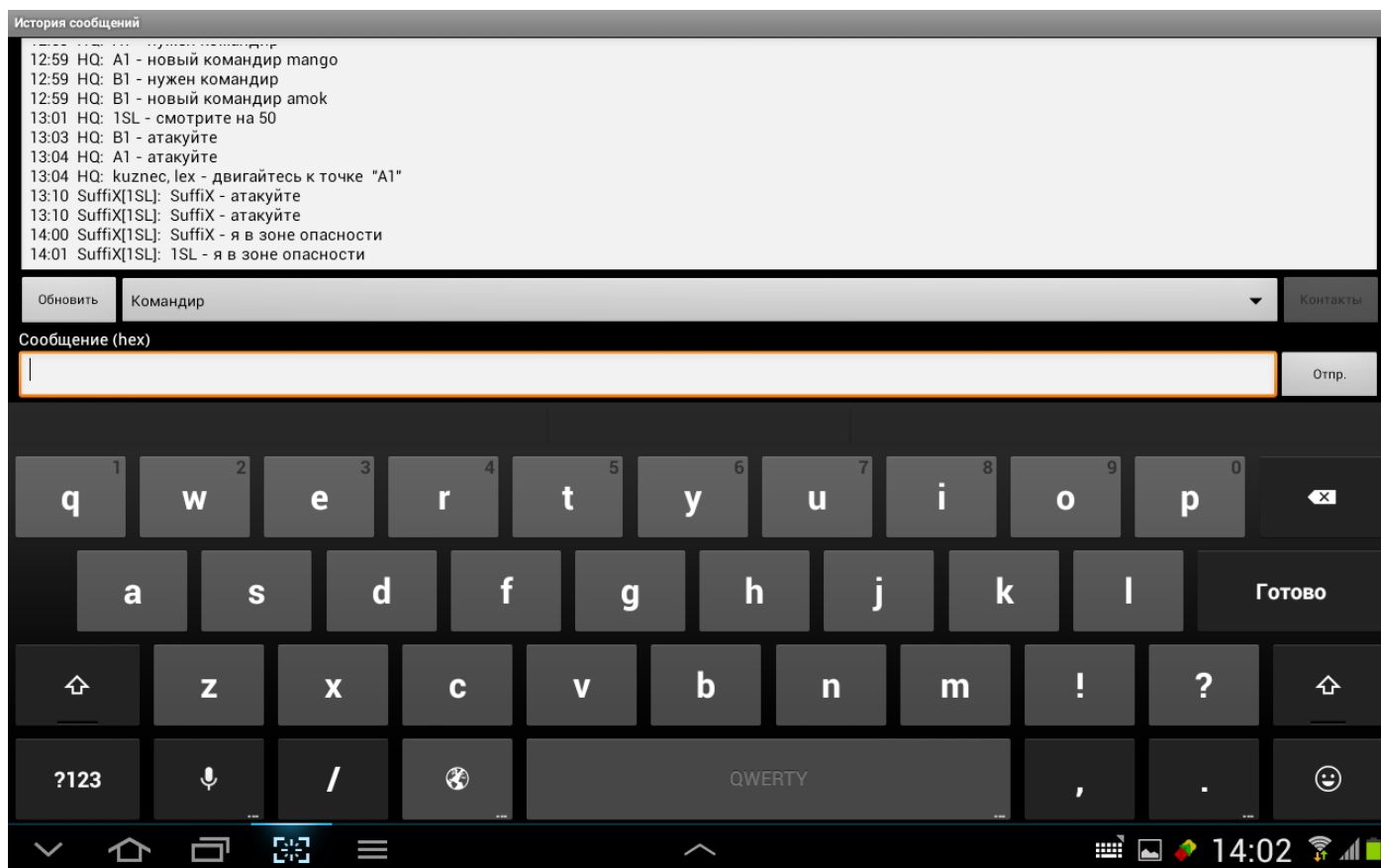


Рисунок 1.13 – Приклад чату між користувачами

Для повноцінної роботи комплексу потрібне інтернет-з'єднання між комплексом і сервером. Однак комплекс також може працювати в автономному режимі як звичайний автономний навігатор без синхронізації даних між користувачами. Інтернет з'єднання в автономному режимі потрібно тільки для завантаження карти з сервера в перший раз. Для повноцінної роботи системи підходить будь-яке інтернет-з'єднання аж до GPRS. Комплекс оптимізовано під роботу на низькошвидкісних і нестабільних з'єднаннях.

Одне з головних переваг - без постійного використання GPS батарея пристрою може працювати 5-8 годин. Користувачі можуть використовувати додаткові акумуляторні батареї або power банки для збільшення часу служби комплексу без підзарядки.

Після пошуку наявних аналогів та їх детального аналізу можливо створити порівняльні таблиці де представлені їх характеристики.

Таблиця 1.1

Порівняльна характеристика звичайних продуктів

Назва продукту	«Где Я»	«Enduro Tracker»	«GPS Лама»
Останнє оновлення	2019	2018	2019
Розробник	KinKin Ltd	Blitzsquad	Levston UK
Тип продукту	Застосунок	Застосунок	Застосунок
Інтерфейс користувача	Задовільний	Перевантажений	Приємний
Функціональність	Можливість обміну повідомленнями, координатами, побудова маршрутів.	Можливість обміну координатами, побудова маршрутів.	Можливість обміну координатами, повідомленнями, побудова маршрутів, кнопка «SOS», створення груп, створення локацій, працює в режимі сну.

Таблиця 1.2

Порівняльна характеристика військових продуктів

Назва продукту	"Базальт-К"	«Андромеда-Д»	«Альянс»	MAP21
Останнє оновлення	2019	2019	2018	2019

Продовження таблиці 1.2

Країна розробник	Україна	Росія	Росія	Україна
Тип продукту	Навігаційний комплекс	Автоматизована система	Застосунок	Застосунок
Інтерфейс користувача	Перевантажений	Перевантажений	Приємний	Приємний
Функціональність	Побудова маршрутів, передача координат, математичні розрахунки з геолокацією, радіообмін	Передача координат, математичні розрахунки з геолокацією, статус об'єктів, радіообмін.	Математичні розрахунки з геолокацією, обмін координатами, можливість побудови маршрутів та обміну повідомленнями.	Математичні розрахунки з геолокацією, обмін координатами, можливість побудови маршрутів та обміну повідомленнями.

Порівнявши дані з таблиць 1.1 та 1.2 можемо отримати необхідні знання для створюваного продукту. Як можливо побачити, розробка нової системи, що буде мати необхідний функціонал якого не достатньо в вже існуючих продуктах та ліпші характеристики є актуальною. Розроблюваний продукт може бути цілком конкурентоспроможний і необхідний для різних сфер життя, як для цивільних, так і для військових потреб.

1.3 Висновок до розділу

У даному розділі було змістовно проаналізовано предметну область, виявлено аналоги, описано предметне середовище, що дозволяє чітко простежити всі особливості предметної області.

Також, було проведено аналіз аналогів розроблюваної системи, який включає короткий опис кожної з систем, а також наведені основні причини постановки рішення про розробку нового програмного забезпечення .

2 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

2.1 Функціональні та не функціональні вимоги

Надалі з'являється необхідність побудувати функціональну модель роботи системи що розробляється. Для цього використаємо діаграму використання (рис. 2.1) та додаток А .

Для вдалого проектування даної діаграми спочатку необхідно визначити дійових осіб (акторів), а потім визначити, які дії може виконувати актор. Актором даної системи є адміністратор групи, не авторизований та авторизований користувач.

Користувач системи – будь-який користувач, який використовує систему для координації наземних рухомих об'єктів.

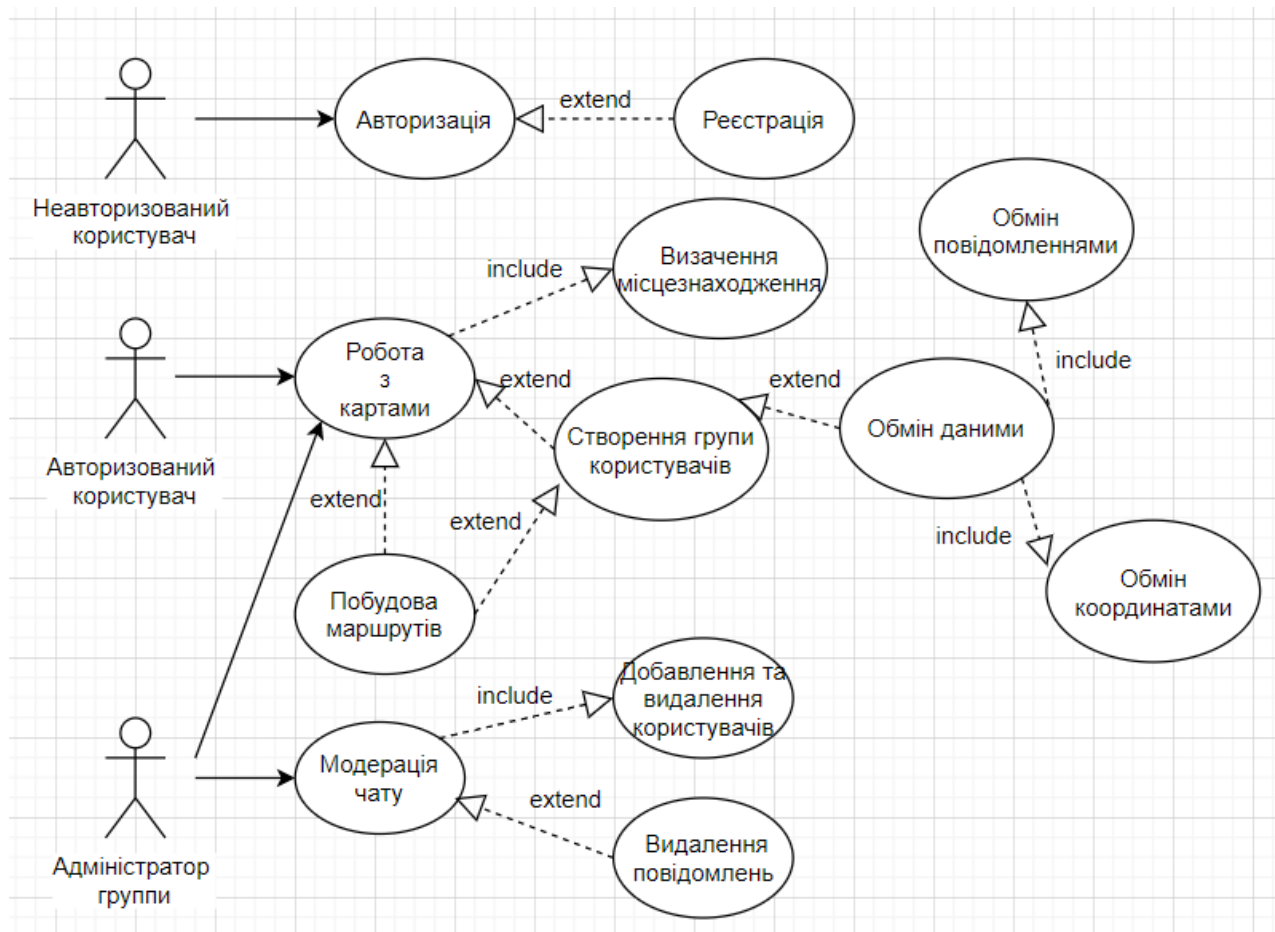


Рисунок 2.1– Діаграма використання

У таблиці 2.1 наведена специфікація функціональних вимог до системи.

Таблиця 2.1

Функціональні вимоги до системи

№	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Актор
1	Реєстрація	Обов'язкове	Низька	Неавторизований користувач
2	Авторизація	Обов'язкове	Низька	Неавторизований користувач
3	Вхід на сторінку з картою	Обов'язкове	Низька	Користувач
4	Обмін даними	Обов'язкове	Висока	Користувач
5	Створення або вхід в групу	Обов'язкове	Середня	Користувач
6	Модерація чату	Середній	Середня	Адміністратор групи

Реєстрація користувача - процедура реєстрації користувача на веб-сайті, інтернет-сервісі з використанням облікових записів. Обліковий запис - набір даних користувача, що зберігаються в комп'ютерній системі, необхідний для його розпізнавання (аутентифікації) та забезпечення доступу до його персональних даних та налаштувань. Для використання облікового запису (тобто для входу під іменем) зазвичай потрібно ввести ім'я (логін) та пароль. Також може знадобитися інша додаткова інформація. Користувачі Інтернету можуть сприймати обліковий запис як особисту сторінку, профіль, акаунт, місце зберігання особистої та іншої інформації, користувачі повідомляють про себе на певному Інтернет-ресурсі.

Авторизація. Для коректної роботи продукти користувач повинен увійти в систему.

Робота з картами. Без визначення місцезнаходження неможлива робота з картами й подальше коректне функціонування системи.

Створення групи. Можливість користувача створити групу із людей, з якими він зможе обмінюватись даними, повідомленнями, координувати свої дії.

Обмін даними. Під цим пунктом мається на увазі обмін між користувачами групи даними з бази даних, обмін повідомленнями .

У таблиці 2.2 наведена специфікація нефункціональних вимог до системи.

Таблиця 2.2

Нефункціональні вимоги до системи

№	Назва вимоги	Атрибути вимог	
		Пріоритет	Складність
1	Багатоплатформеність	Високий	Низька
2	Доступність для нових користувачів	Високий	Низька
3	Легкість інтерфейсу	Середній	Середня
4	Безпека	Високий	Висока
5	Надійність	Середній	Висока
6	Відмовостійкість	Низький	Висока
7	Цілісність даних	Високий	Висока
8	Керованість	Середній	Середня
9	Час відновлення	Середній	Середня

Багатоплатформенність або міжплатформенність - можливість програмного забезпечення працювати з двома або більше апаратними платформами та (або) операційними системами. Вона надається за допомогою використання мов програмування високого рівня, середовищ розробки та режиму виконання, а також підтримує умови компіляції, зв'язування та виконання коду для різних платформ. Типовим прикладом є програмне забезпечення, розроблене для роботи в операційних системах Linux та Windows одночасно.

Багатоплатформенною можна назвати більшість сучасних мов програмування високого рівня. Наприклад, C, C ++, Free Pascal, FreeBASIC, PureBasic - багатоплатформна мова на рівні компіляції, тобто для цих мов є компілятори для різних платформ. Це дозволяє - при належній якості коду - не переписувати основний двигун програми, змінюючи лише конкретні залежні від системи частини.

Доступність для нових користувачів пояснюється тим, що звичайний користувач, який вперше перейшов до продукту, що розробляється, міг негайно зрозуміти системні модулі та як правильно користуватися системою. Наявність - це час, протягом якого новий користувач отримує доступ до послуги. Доступність визначається декількома факторами, найважливішими з яких є надійність системи, її продуктивність та кількість користувачів.

Легкість інтерфейсу можна розділити на кілька категорій, які перераховані нижче.

1. Продуктивність. Інтерфейс це - інструмент управління, який надає доступ користувачу до всіх функцій веб-сайту. І цей доступ для користувача повинен бути максимально зручним. Для цього інтерфейс необхідно створити таким, щоб будь-яку інформацію або необхідні функції людина могла знаходити відразу, що зробить його роботу з проектом комфортною.
2. Відповідність контексту. Розробляючи дизайн інтерфейсу, необхідно враховувати тип вмісту сторінки. Різні сторінки не повинні мати різний зміст. Потрібно створити

такі елементи управління сайтом, які зроблять роботу користувача простою та зручною, а потім приведуть її до автоматичності.

3. Авторитетність. Створюючи інтерфейс, ви не повинні покладатися на «інтуїтивну поняття» його користувачем.

4. Захоплення. Інтерфейс не можна вважати хорошим, якщо він не звертає на себе увагу. Можна працювати на простому та зручному, ефективному та оперативному інтерфейсі, але лише привабливість перетворить роботу користувача у справжнє задоволення. А оскільки " про смаки не сперечаються ", і догодити всім-дуже складне заняття, при створенні свого продукту потрібно орієнтуватися на цільову аудиторію та її переваги.

5. Уважність. У цьому випадку наголошується на увагу до користувача. Швидкість інтерфейсу сайту повинна бути високою. Користувач буде роздратований, коли потрібно довго чекати завантаження сторінки. Обов'язково потрібно встановлювати зворотній зв'язок з користувачами, систему сповіщень про те, що відбувається на сайті, підтвердження дій, які виконуються користувачем.

6. Мінімалізм. Нагромодження інтерфейсу піктограмами, переходами та іншими елементами управління не принесе нічого доброго. Доступність у цьому випадку буде лише очевидною, але насправді це дасть змогу користувачеві надовго зрозуміти, для чого він існує і куди це призведе.

7. Чіткість. Для дизайну будь-якого інтерфейсу однією з головних властивостей є зрозумілість та доступність використання. Взаємодія з системою повинна бути легкою для будь-якого користувача. Якщо людині незрозуміла робота додатку, він, швидше за все, просто відмовиться від нього. Отже, будь-який веб-сайт або додаток повинні бути інтуїтивно зрозумілими для користувача.

Надійність – частка часу неперервної роботи системи. Чим вище це значення, тим менше система простоє. Для дуже важливих застосунків слід досягти принаймні 99,9% надійності. Загальна вимога сьогодні - «п'ять дев'яток» - п'ять

хвилин простою на рік. Ті ж вимоги повинні бути встановлені перед мережевим обладнанням, каналами зв'язку та джерелами живлення. Надійність серверів повинна бути вищою, ніж надійність роботи та мобільних пристроїв.

Цілісність даних. Дані, які знаходяться у правильно працюючій системі, повинні бути послідовними та адекватними (відповідати дійсності). Система стає нефункціональною, якщо до неї внесені несанкціоновані зміни. Перевірити точність даних досить складно, однак у критичних інформаційних системах необхідно забезпечити контроль цілісності, наприклад, контрольною сумою або за допомогою електронного цифрового підпису.

Відмовостійкість – кількість одночасних відмов компонентів системи, що викликають відключення всієї системи. Чим більше системних вузлів потрібно відключити, щоб зупинити його, тим більше система є відмовостійкою. Відмовостійкість підвищує загальну надійність системи, що складається з недостатньо надійних компонентів. Вимоги до відмов визначаються різницею між необхідним рівнем надійності та фактичною надійністю існуючих компонентів.

Час відновлення. Навіть найнадійніші системи іноді виходять з ладу. Час відновлення після аварійного вимкнення або відключення системи також є важливою вимогою. Більше того, відновлення системи - це не лише надання доступу до серверів та додатків, а й відновлення цілісності даних.

Керованість. Для зміни конфігурацій та виправлення помилок у системі слід передбачити механізми управління її станом. Якщо система виходить з-під контролю адміністратора, то не можна говорити про її регулярну безперервну роботу, оскільки вона може неадекватно реагувати на запити користувачів. Тому бажано, щоб система мала механізми моніторингу її стану та інструменти для самоконфігурації без зупинки всієї системи.

Безпека. Що стосується несанкціонованих змін у системі та порушення цілісності даних, потрібно передбачити механізми ідентифікації і авторизації користувачів, а також забезпечити конфіденційність спеціальних даних. В деяких випадках

порушення конфіденційності інформації еквівалентна виходу системи із ладу. Це, наприклад відноситься до даних системи контролю цілісності. Тому, потрібно передбачити механізми ідентифікації користувачів, авторизації їх доступу до ресурсів системи і криптографічного захисту конфіденційних даних.

2.2 Бізнес правила

Правила ведення бізнесу - це особливий тип логіки, який описує обмеження типу дій, які система чи люди повинні враховувати у своїй поведінці. Ці правила визначаються різними чинниками, включаючи нормативні настанови, галузеві стандарти, проникливість бізнесу та здоровий глузд. Вони часто відрізняються від країни до країни, від галузі до галузі і навіть від компанії до компанії. Прикладом правила банківського бізнесу є закон, який вимагає повідомляти про угоди з грошовою сумою понад 10 000 доларів США. Без сумніву, це правило бізнесу потрібно враховувати при створенні банківської системи депозиту / зняття готівки.

Правила бізнесу існують на різних рівнях. Деякі з них впливають на всю систему, а багато систем справді створені для забезпечення ділових правил. Правила ведення бізнесу також можуть сильно відрізнятися за сферою застосування. Тим не менш, правила ведення бізнесу мають одне спільне: вони керують частиною бізнесу. За визначенням, ділове правило - це обмеження, яке стосується людини. Далі пояснюються деякі деталі, характерні для уніфікованих мов моделювання.

Зазвичай існує три типи правил які наведені нижче.

Правило деривації (dereviation rule) перетворює отриману інформацію у повернені значення. Наприклад, знижки на продукцію можна розраховувати за допомогою спеціального алгоритму, який враховує розмір замовлення, рекламну підтримку та важливість замовника, якому товар доставлений. Правила такого типу допускають зміни, тому їх потрібно висвітлити, перш ніж їх можна буде обробити.

Правило обмеження (constraint rule) перевіряє цінність транзакції або операції на відповідність та не суперечливість. Наприклад, щоб клієнт отримав лист, поштовий

індекс на листі повинен відповідати країні проживання замовника. Ці правила можна використовувати для дослідження взаємозв'язків між об'єктами. Наприклад, коли всі клієнти мають звертатися до об'єктів в інтернет-магазині обладнання. Вони також можуть використовуватися в поєднанні з результатами шаблону. Якщо вони не щирі, ми не можемо продовжувати або завершувати процес.

Нарешті, існують інваріантні правила. Інваріантні правила (invariant rules) перевіряють наявність численних змін та забезпечують узгодженість результатів. Наприклад, залишок на рахунку повинен бути попереднім залишком плюс доходи або витрати. Якщо ви не згодні, ваша система втрачає гроші і настав час підняти виняток на більш високий рівень.

Бізнес-правила в базі даних слугують умовою відповідності даних предметній області(рис. 2.2)[10].

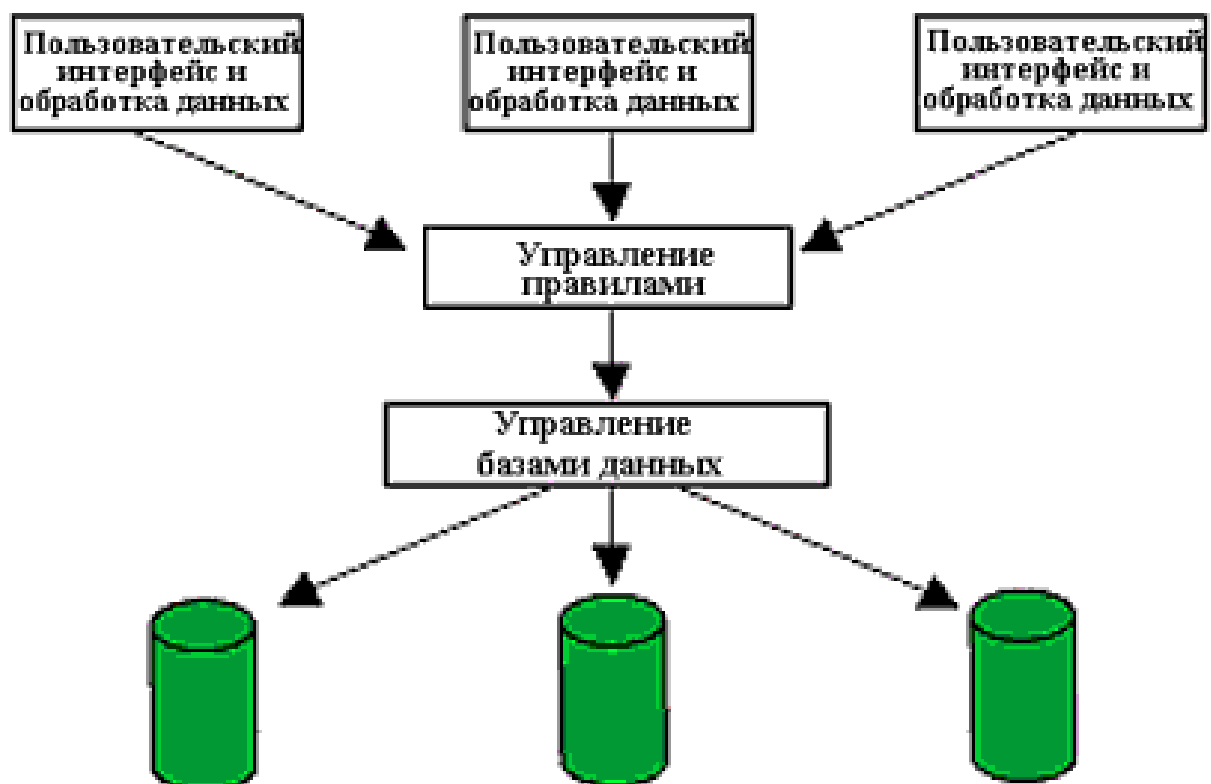


Рисунок 2.2 – Приклад системи бізнес правил

Бізнес-правила можна розділити на елементарні і розширені:

- елементарні правила обмежують значення певного атрибута або сукупності атрибутів через обмеження домену;
- розширені правила виражаються як залежність між атрибутами.

Обмеження можуть бути статичними або динамічними. Статичні обмеження повинні бути дотримані для кожного стану бази даних. Динамічні обмеження очевидні при переміщенні бази даних з одного стану в інший. Наприклад, якщо зарплата працівника зростає, нове значення повинно бути більшим ніж старе.

Обмеження можуть знайти своє вираження:

- при описі атрибутів відносин в концептуальній схемі;
- в запиті до бази даних;
- в процедурі бази даних;
- в правилі (в тригері).

Процедури бази даних створюються проектувальником бази даних і доповнюють СУБД. У різних СУБД їх називають збереженими, приєднаними, спільними тощо. У системах з архітектурою клієнт-сервер використання процедур бази даних може значно зменшити мережевий трафік. Додаток, який викликає процедуру, передає серверу лише ім'я та параметри.

Правило (тригер) - програма для моніторингу ситуації, яка виникає при внесенні змін до бази даних. Правило передбачено для таблиці бази даних і використовується при виконанні операцій для включення, видалення або відновлення рядків та при зміні значень у стовпцях таблиці.

Для організації роботи системи координації наземних рухомих об'єктів потрібно виконати такі основні бізнес-процеси.(рис. 2.3) та додаток Б:

- процес роботи з картою;
- процес обміну повідомленнями;
- процес модерації;
- процес координації об'єктів.

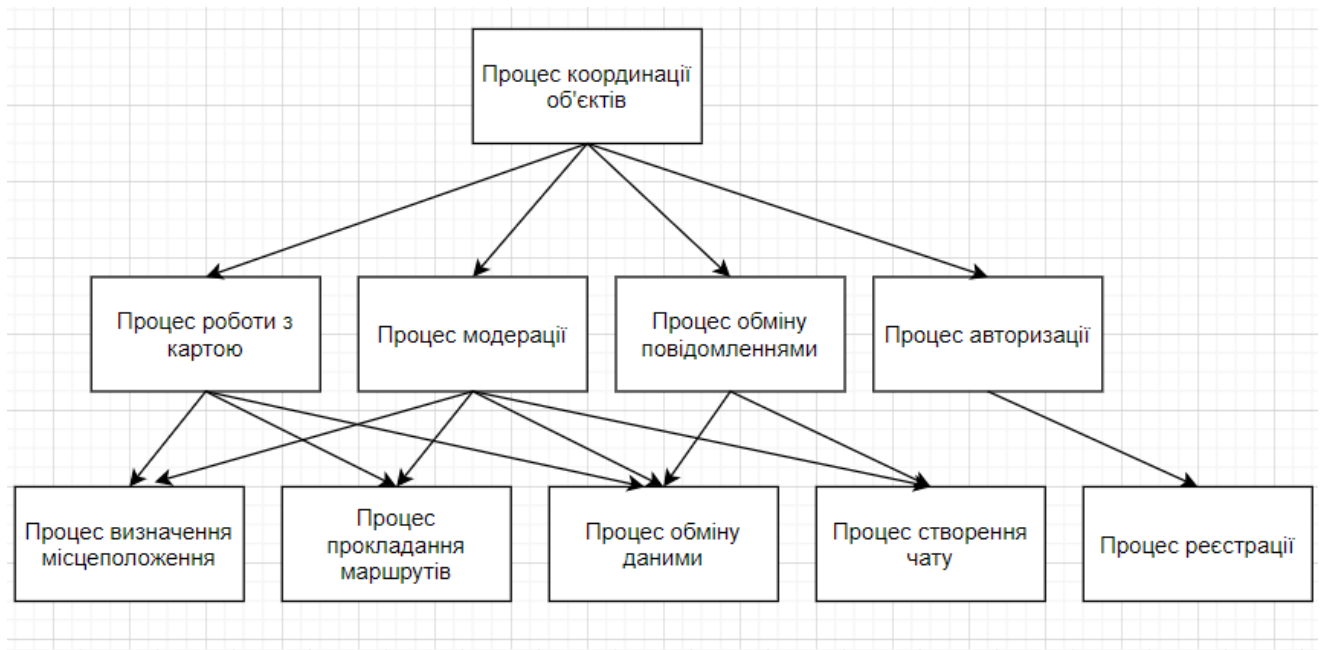


Рисунок 2.3 – Діаграма дерево функцій

У таблицях 2.3-2.6 знаходиться детальна характеристика основних бізнес процесів.

Таблиця 2.3

Характеристика процесу роботи з картою

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Робота з картою
Актор	Користувач
Вхідна подія	Вхід на головну сторінку
Вхідні дані	Дані про авторизованого користувача
Вихідні дані	Точні координати користувача
Вихідна подія	Відображення координат
Процеси-клієнти	Процес визначення місцезнаходження, процес прокладання маршрутів, процес обміну даними

Таблиця 2.4

Характеристика процесу модерації

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Модерація
Актор	Адміністратор
Вхідна подія	Вхід на головну сторінку як адміністратор
Вхідні дані	Дані про адміністратора
Вихідні дані	Точні координати користувача
Вихідна подія	Відображення координат, можливість адмініструвати налаштування
Процеси-клієнти	Процес визначення місцеположення, процес прокладання маршрутів, процес обміну даними, процес створення чату

Таблиця 2.5

Характеристика процесу обміну повідомленнями

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Обмін повідомленнями
Актор	Користувач
Вхідна подія	Запит на створення чату
Вхідні дані	-
Вихідні дані	Створена група
Вихідна подія	Відображення групи користувачів
Процеси-клієнти	Процес обміну даними, процес створення чату

Таблиця 2.6

Характеристика процесу обміну авторизації

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Обмін повідомленнями
Актор	Не авторизований користувач
Вхідна подія	Вхід на сторінку авторизації
Вхідні дані	-
Вихідні дані	-
Вихідна подія	Перехід на головну сторінку
Процеси-клієнти	Процес реєстрації

2.3 Цілі та задачі розробки

Мета дисертації - розробка системи для полегшення процесу координації наземних рухомих об'єктів та створення суспільно-корисного продукту з більшим функціоналом та ліпшими характеристиками.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- аналіз та вивчення теоретичної частини;
- опис архітектури майбутнього виробу;
- створення скрипту геолокації;
- створення чату для обміну і координації ;
- підключення бази даних;
- відображення інформації у графічному вигляді.

Створення скрипту. Данний скрипт дає нам можливість працювати з онлайн картою. Створення чату. Чат допоможе користувачам координувати свої переміщення і просто спілкуватися. Підключення бази даних. База даних, в якій будуть зберігатися дані про користувачів, id кожної групи.

Методи дослідження системи координації наземних рухомих об'єктів:

- дослідження архітектури систем управління об'єктами відбувалося за допомогою аналізу наявних аналогів, завдяки якому ми визначили потрібну архітектуру майбутнього виробу;
- дослідження функцій даних систем відбувалося за допомогою узагальнення різних супутникових та радіо систем, завдяки яким ми дізналися й які саме функції буде виконувати майбутній виріб.

2.4 Опис процесу діяльності

Процес діяльності відбувається дуже просто, його показано в додатку В. На рисунку 2.4 зображена діяльність входу в систему.

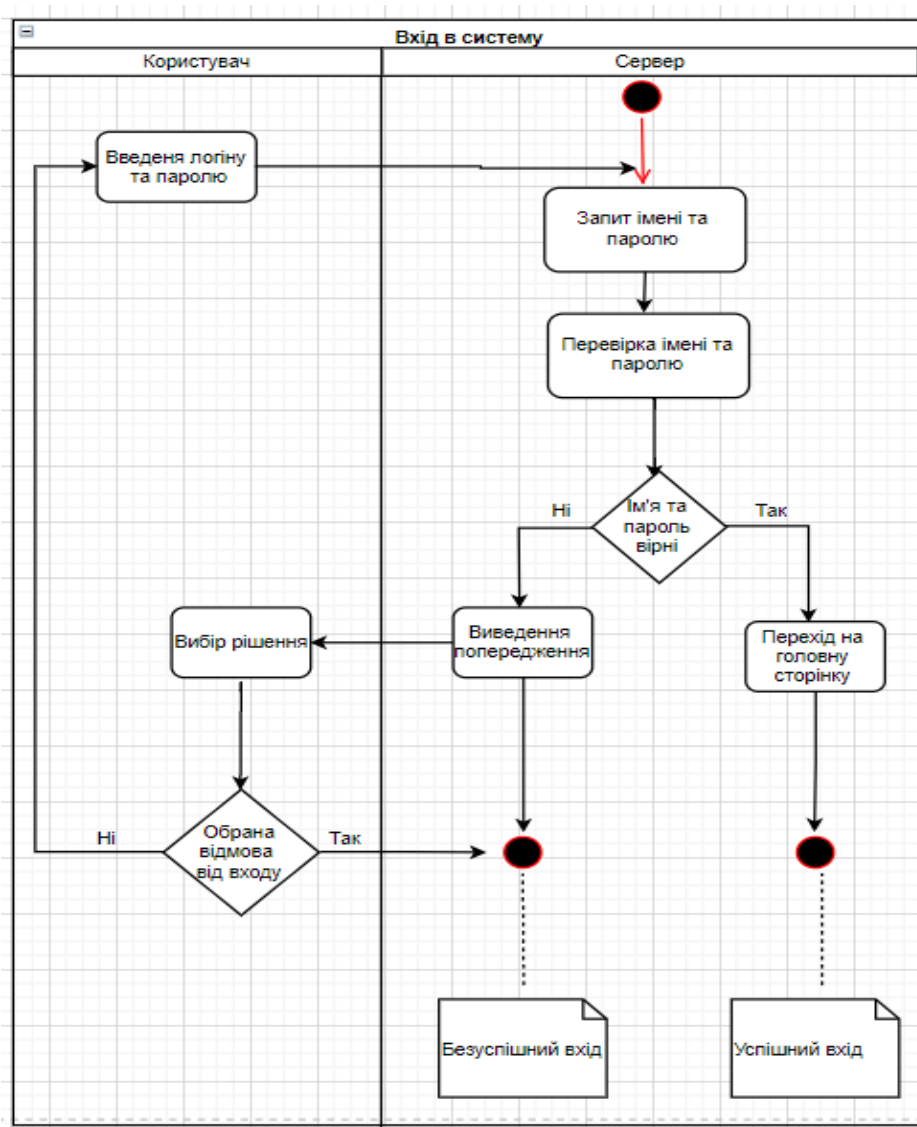


Рисунок 2.4 –Діаграма діяльності входу в систему

Коли користувач заходить на сторінку розроблюваного продукту він повинен пройти авторизацію за допомогою Google аккаунта, або, у випадку відсутності даного, пройти реєстрацію.

Далі користувач переходить на сторінку з картою, де він повинен дозволити використання свого місцезнаходження(рис. 2.5). Далі користувач має можливість створити групу з людей, з якими він матиме змогу ділитися своїм місцезнаходженням, обмінюватись повідомленнями, будувати маршрути. В подальшій розробці системи буде використано більше функціоналу та забезпечення безпеки системи.

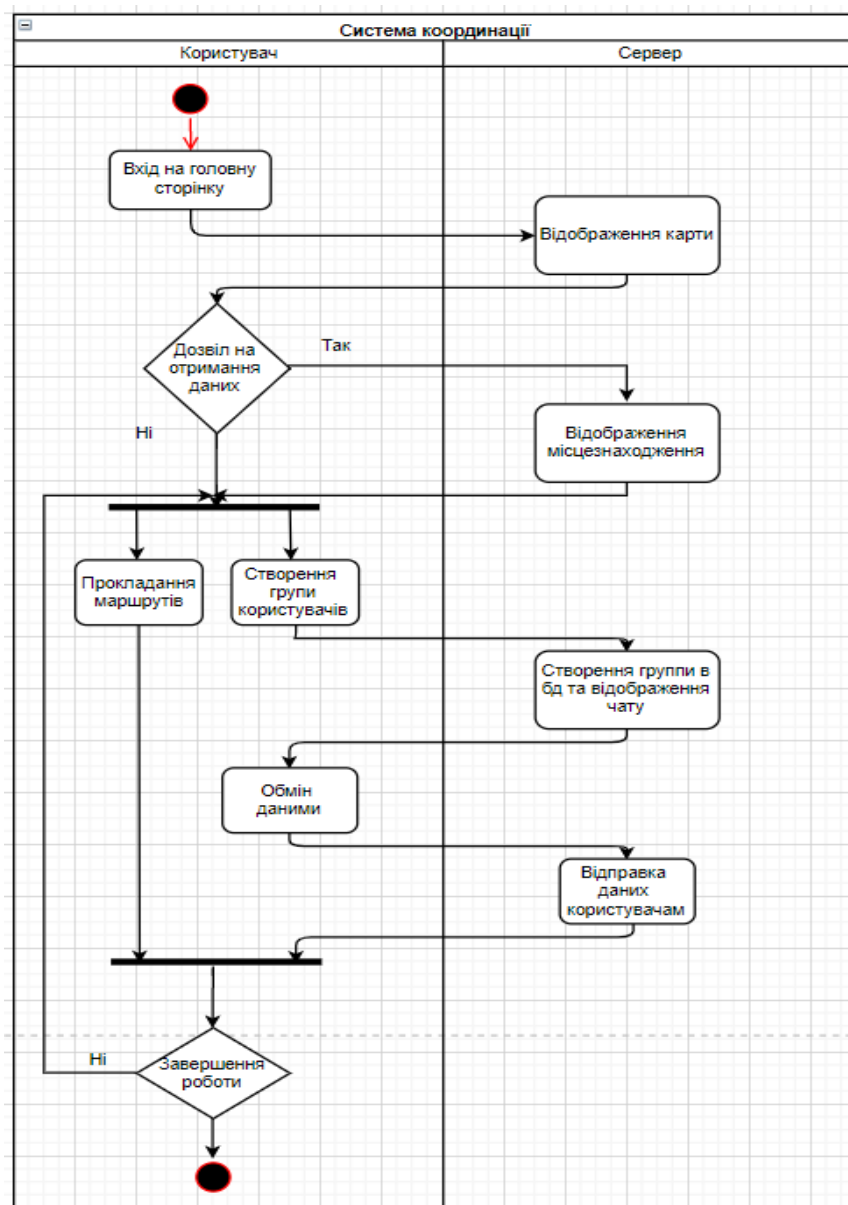


Рисунок 2.5 –Діаграма діяльності системи координації

2.5 Структурна схема

На початку розробки проекту виникає необхідність у побудові структурної схеми для кращого розуміння системи і правильного розбиття системи на блоки.

Структурна схема - схема, яка визначає основні функціональні частини виробу, їх взаємозв'язки та призначення. Під функціональною частиною розуміють складову частину схеми: елемент, пристрій, функціональну групу, функціональну ланку.

Структурна схема розроблюваної системи наведена на рисунку 3.1 та додатку Г.



Рисунок 3.1 – Структурна схема системи координації наземних рухомих об'єктів за допомогою GPS

Робота користувача розпочинається після успішного проходження авторизації коли користувач переходить на головну сторінку з картою, де він вже матиме змогу побудови маршрутів та визначення місцезнаходження.

Далі користувач матиме можливість створити групу користувачів для зручної координації дій між ними. Користувач, який створив групу стає адміністратором, і

має можливості по добавленню та видаленню користувачів та видаленню повідомлень.

Кожен користувач групи має можливість обміну повідомленнями та даними з іншими користувачами для зручної координації.

2.6 Висновок до розділу

У розділу було визначено і проаналізовано функціональні та не функціональні вимоги, побудовано діаграму використання, структурну схему та визначено дійових осіб.

Також були проаналізовані бізнес правила, побудовано дерево функцій і детально розглянуто їх. Було визначено мету розробки системи, визначені та поставлені цілі та завдання. Було проведено опис процесу діяльності, та методів дослідження.

3 СЦЕНАРІЙ ВИКОРИСТАННЯ СИСТЕМИ

3.1 Основний сценарій використання системи

Випадок використання - це модель людей, які описують взаємодію між користувачами та системою. Перевага випадку використання полягає в тому, що користувачі можуть уявити, як вони можуть виконувати завдання, та визначати функції, які можуть знадобитися для підтримки кожного кроку. Крім того, користувачеві залишається лише думати про те, що настає після кроку, що значно полегшує йому правильне визначення всіх попередніх кроків та всіх наступних кроків, не пропускаючи жодного. Випадки використання описують, що повинен робити користувач, що він хоче, і як система реагує під час використання програмного забезпечення.

Основний сценарій використання системи можливо побачити на діаграмі використання. Розглянемо детальніше варіанти використання.

3.2 Сценарій авторизації користувача

Таблиця 3.1

Назва сценарію	Авторизація
Цілі	Користувач: авторизуватися в системі і почати працювати; Система: ідентифікувати користувача і його права.
Дійові особи	Користувач, система
Опис	Для успішного використання системи користувач повинен пройти авторизацію
Тригер	Користувач заходить на сторінку авторизації
Передумова	Обліковий запис користувача не активний

Продовження таблиці 3.1

Успішний сценарій	<p>Користувач запускає систему. Система відкриває сесію користувача, пропонує ввести логін і пароль.</p> <p>Користувач вводить логін і пароль.</p> <p>Система перевіряє логін і пароль.</p> <p>Система створює запис в історії авторизацій (IP адрес користувача, логін, дата, робоча станція).</p> <p>Система видає користувачеві повідомлення з приводу успішної авторизації</p>
Результат	Користувач переходить на головну сторінку продукту

3.3 Сценарій реєстрації

Таблиця 3.2

Назва сценарію	Реєстрація
Цілі	<p>Користувач: пройти реєстрацію;</p> <p>Система: зареєструвати дані користувача в базі даних</p>
Дійові особи	Користувач, система
Опис	-
Тригер	Користувач натискає кнопку реєстрації
Передумова	Користувача немає в базі даних
Успішний сценарій	Користувач вводить свої дані, система зберігає дані користувача
Результат	Система переадресує користувача на сторінку авторизації

3.4 Сценарій входу на сторінку за картою

Таблиця 3.3

Назва сценарію	Вхід до сторінки з картою
Цілі	Користувач: почати роботу з картою; Система: графічно відобразити головну сторінку
Дійові особи	Користувач, система
Опис	-
Тригер	-
Передумова	-
Успішний сценарій	Користувач зайшов на сторінку , система відобразила сторінку
Результат	Користувач може почати роботу з картою

3.5 Сценарій визначення місцезнаходження

Таблиця 3.4

Назва сценарію	Визначення місцезнаходження
Цілі	Користувач: почати роботу з картою; Система: визначити координати користувача і відобразити їх на карті
Дійові особи	Користувач, система
Опис	Для роботи з картою спочатку система повинна знати місцезнаходження користувача
Тригер	-
Передумова	-
Успішний сценарій	Користувач дає дозвіл системі отримати дані про своє місцезнаходження, система отримує дані, записує їх в базу даних і відображає користувачу.

Продовження таблиці 3.4

Результат	Користувач може продовжити роботу з картами або обрати інші операції
-----------	--

3.6 Сценарій побудови маршрутів

Таблиця 3.5

Назва сценарію	Побудова маршрутів
Цілі	Користувач: прокласти маршрут; Система: графічно відобразити прокладений користувачем маршрут
Дійові особи	Користувач, система
Опис	-
Тригер	Користувач натискає кнопку прокладання маршрутів
Передумова	-
Успішний сценарій	Користувач ввів адреса або поставив точки на карті, система проклала маршрут і відобразила його на карті
Результат	Користувач може використати прокладений маршрут

3.7 Сценарій створення групи користувачів

Таблиця 3.6

Назва сценарію	Створення групи користувачів
Цілі	Користувач: створити групу користувачів; Система: графічно відобразити чат, зареєструвати

Продовження таблиці 3.6

	групу в базі даних
Дійові особи	Користувач, система
Опис	-
Тригер	Користувач натискає кнопку створення групи
Передумова	Користувач може створити групу
Успішний сценарій	Користувач натискає кнопку створення групи, система створює групу, привласнює їй номер, графічно відображає користувачу поля, де користувач вводить id або логін іншого(-их) користувача(-ів), натискає кнопку додати. Система додає інших користувачів, відображає чат.
Результат	Користувачі можуть обмінюватись даними в групі

3.8 Сценарій обміну даними

Таблиця 3.7

Назва сценарію	Обмін даними
Цілі	Користувач та адміністратор: обмінюватись координатами і повідомленнями; Система: забезпечити обмін даними між користувачами
Дійові особи	Користувач, система, адміністратор
Опис	-
Тригер	-
Передумова	Наявність більше 1 користувача в чаті
Успішний сценарій	Користувачі успішно обмінюються повідомленнями в чаті
Результат	-

3.9 Сценарій додавання та видалення користувачів

Таблиця 3.8

Назва сценарію	Додавання та видалення користувачів
Цілі	Адміністратор: змінювати склад учасників групи; Система: забезпечити додавання та видалення користувачів в базі даних групи.
Дійові особи	Система, адміністратор
Опис	-
Тригер	Адміністратор натискає кнопку додавання/видалення користувачів
Передумова	Користувач є адміністратором
Успішний сценарій	Система відображає поле для id або логіну користувача, якого адміністратор хоче додати/видалити, адміністратор вводить дані, натискає кнопку «готово», система знаходить такого користувача і додає/видаляє його
Результат	Зміна кількості учасників в групі

3.10 Сценарій видалення повідомлень

Таблиця 3.9

Назва сценарію	Видалення повідомлень
Цілі	Адміністратор: видалення небажаних повідомлень; Система: забезпечення коректного видалення повідомлення для всіх користувачів.
Дійові особи	Система, адміністратор
Опис	-

Продовження таблиці 3.9

Тригер	Адміністратор натискає кнопку видалення повідомлень
Передумова	Користувач є адміністратором
Успішний сценарій	Адміністратор обирає які повідомлення він хоче видалити, натискає кнопку видалення, система видаляє їх.
Результат	Повідомлення видалені

3.11 Висновок до розділу

В розділу був проведений аналіз основного сценарію використання системи, а також детально розглянуто кожен прецедент, що допомогло отримати потрібні знання для побудови розроблюваного продукту.

4 ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ

4.1 Мови програмування

Для розробки системи для координації об'єктів було обрано такі мови програмування: на front-end – JavaScript, CSS, HTML , а на back-end – Java.

4.1.1 JavaScript

JavaScript – мультіпарадигменна мова програмування. Вона підтримує об'єктно-орієнтований, імперативний і функціональний стилі. Є реалізацією мови ECMAScript .

Браузеру користувача потрібні три речі, щоб користувач міг користуватись веб-сторінками:

- HTML структурує вміст сторінки;
- CSS стилізує сторінку;
- JavaScript змушує код працювати правильно.
- Це робить все це прямо, але, повірте, це є набагато більше

Початкова мета мови програмування JavaScript полягала у тому, щоб перетворити Інтернет у платформу додатків, тобто JavaScript працюватиме як на клієнті, так і на сервері. JavaScript потребувало багато років, щоб сприйматись серйозною мовою. Але вона швидко розповсюдилась на front-end, ставши стандартною мовою програмування в Інтернеті.

JavaScript є об'єктно-орієнтованою, але той факт, що в мові використовується прототипування викликає відмінності в роботі з об'єктами в порівнянні з традиційними мовами що орієнтуються на клас .

Не дивлячись на той факт, що JavaScript є об'єктно-орієнтованою, вона має ряд властивостей, які мають функціональні мови – каррінг, замикання, функції як

об'єкти першого класу, об'єкти як списки, анонімні функції – що наділяє мову додатковою гнучкістю.[11]

Незважаючи на подібний синтаксис з мовою «С», JavaScript має принципові відмінності:

- автоматичне прибирання сміття;
- об'єкти з можливістю інтроспекції;
- автоматичне приведення типів;
- анонімні функції;
- функції як об'єкти першого класу.

Переваги JavaScript:

- легкість входження й розуміння мови;
- жоден сучасний браузер не обходиться без підтримки JavaScript;
- виконання логіки на стороні клієнта приносить швидший досвід користувачам(код працює безпосередньо в браузері, потреба у викликах сервера обмежується, отже, скорочення часу завантаження);
- з самого початку JavaScript впроваджував інтерактивність користувацького інтерфейсу в Інтернет;
- мова постійно удосконалюється ;
- JavaScript стоїть за будь-яким хорошим веб-дизайном;
- працювати з сторінками можна навіть із текстових редакторів;
- перспектива використання мови в процесі навчання програмуванню і інформатиці.

Недоліки JavaScript:

- JavaScript має низький рівень безпеки через відкритість і розповсюдженість коду та популярних скриптів ;

- тисячі пакетів, що складають екосистему JS, дозволяють розробникам швидко працювати, не винаходити колесо для кожного нового завдання, однак вони також спричиняють те, що дехто називав "пекло залежності";
- злиття ваших проектів за допомогою JS надасть вам шкоду в довгостроковій перспективі, оскільки це врешті-решт спричинить проблеми з ефективністю
- велика частина програм які наразі активно використовуються при відсутності мови просто перестануть існувати.

4.1.2 CSS

CSS – це те, що входить у формат макета веб-сайту. Це набір правил форматування, які допомагають розробникам веб-сайтів контролювати зовнішній вигляд та показ веб-сайту, над яким вони працюють. CSS дозволяє їм визначати стильові рішення, такі як розташування зображень, розмір шрифту, колір тла та все інше, що впливає на те, як веб-сайт відображатиметься у вашому веб-браузері. Крім того, за допомогою онлайн-редактора ви можете впорядкувати, налагодити свої стильові таблиці та легко стиснути CSS. CSS має чудову підтримку в сучасних браузерах. Вивчити CSS і створити кращий і чистіший код легше, ніж застосовувати всі ці стилі безпосередньо до свого HTML.

CSS використовують для завдання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду на веб-сторінках [12].

CSS - це функціональна мова, яка дозволяє реагувати на веб-сайті. Її використання - це те, що реально визначає переваги та недоліки CSS.

Переваги CSS:

- покращує швидкість завантаження сторінки на веб-сайті, адже чим більше код, тим повільніше стає веб-сайт, а відвідувачі швидко відмовляються від веб-сайту, якщо на це потрібно більше 2-3 секунд;
- всього кілька рядків коду - це все, що потрібно для внесення змін до великої кількості сторінок на веб-сайті за допомогою CSS, а оскільки код мінімальний,

база даних веб-сайту залишається незабрудненою, таким чином, усуваючи будь-які проблеми із завантаженням веб-сайтів;

- за допомогою єдиного аркуша стилів ви зможете переконатися, що зміни виглядають рівномірно на всіх сторінках, якби не CSS, користувачу доведеться робити нотатки про зміни, внесені на одну сторінку, і посилатись на неї під час внесення змін на іншу сторінку, завжди повертаючись вперед і назад;
- допомагає пошуковій системі краще сканувати ваші веб-сторінки, адже боти пошукової системи дають неточні деталі коли вони проходять через громіздку купу HTML-коду, але коли діє CSS, атрибути дизайну веб-сайтів визначені, і на ньому менше коду, що робить веб-сайт більш SEO дружнім;
- можливість перестановки CSS дозволяє визначати зміни в положенні веб-елементів, присутніх на сторінці, а саме: розробники можуть розміщувати елементи HTML у потрібному місці з метою узгодження з естетичною привабливістю сторінки чи іншими подібними міркуваннями;
- зміни CSS для сумісності пристроїв, оскільки для доступу до веб-сайтів в Інтернеті користуються великою кількістю різноманітних розумних пристроїв, існує потреба у багатоплатформенному веб-дизайні, а CSS служить цій меті, роблячи ваші веб-сторінки багатоплатформеними, щоб вони відображалися однаково на всіх пристроях;
- CSS використовує web-стандарти, що і служить підвищенню сумісності з різними платформами.

Недоліки CSS:

- різне відображення верстки в різних браузерах: якщо браузери застарілі, то можливо, що одні й ті ж дані CSS по-різному ними інтерпретуються;
- світ мови програмування сам по собі надзвичайно складний для не розробників та новачків, щоб додати до CSS різні рівні, тобто CSS, CSS 2, CSS 3, потрібно знати дуже багато того, що може заплутати недовсвідченого користувача;

4.1.3 HTML

У епоху ІТ-світу, що швидко змінюється, дуже важливо, як виглядає веб-додаток (настільний або мобільний) будь-якої компанії. Оскільки високий попит на інновації та креативність на веб-сайтах, набагато більше, ніж просто текст та інформація, на ринку з'явилися численні мови, що надають різні можливості розробникам програм для розробки програми.

HTML розшифровується як мова програмування гіпертексту, яку створив Бернерс Лі. На сьогоднішній день HTML використовується з іншими мовами, такими як Javascript, CSS, що надає більше вигляду та відчуття веб-сайту, надаючи різний колір шрифту, розмір шрифту та вирівнювання всього вмісту.

HTML - це один з найкращих варіантів розробити веб-сторінку чи веб-сайт для малого чи зростаючого бізнесу, який не хоче більше інвестувати у придбання програмного забезпечення або її ліцензію і не вимагає розширеного програмування для своїх веб-сайтів.

Нижче наведені переваги HTML [13].

1. HTML легко вивчити та використовувати. Для того, хто навчається веб-розробці, HTML - це перша і головна мова, через яку людина перейде. У ньому є прості теги, і в HTML не існує суттєвої чутливості регістру. У ньому просто є кілька тегів, які відповідають певній меті, і це все. Можна легко зрозуміти чужий код і за потреби вносити зміни до нього, оскільки в ньому немає багато іншого для розуміння. Більше того, він не кидає помилок і не створює жодних проблем, як інші мови програмування, якщо розробник забуде закрити теги або допустити деякі помилки в коді.
2. Однією з найбільших переваг HTML є те, що він безкоштовний і немає необхідності купувати конкретне програмне забезпечення. Не слід мати справу з різними плагінами, необхідними для роботи над будь-яким програмним забезпеченням, оскільки для HTML не потрібні додатки. Таким

- чином, це дуже економічно вигідно з точки зору бізнесу, оскільки немає вартості придбання ліцензії, якщо весь веб-сайт розроблений мовою HTML.
3. Практично всі браузерери по всьому світу підтримуються HTML. Тому не потрібно турбуватися про веб-сайт, написаний в HTML для підтримки браузера, оскільки веб-сайт легко відображатиметься у всіх браузерах, якщо програма має на увазі оптимізувати веб-сайт для різних браузерів. HTML надає простий спосіб оптимізувати веб-сайт у HTML відповідно до браузерів веб-розробникам.
 4. HTML - це одна з найдружніших пошукових систем порівняно з усіма мовами програмування, доступними на ринку (дружній для пошукових систем означає надання якісних веб-сайтів користувачам з відповідною інформацією під час пошуку конкретного веб-сайту). Досить простіше створити веб-сайти, сумісні з SEO, використовуючи HTML, ніж інші мови програмування. Веб-сканери веб-сайтів HTML легше читати і отримувати доступ до них, це скорочує час розбору та час завантаження сторінки на веб-сайті, а отже, покращує його ефективність.
 5. HTML можна легко інтегрувати з декількома мовами і не створювати в них жодних проблем. Наприклад, у Javascript, Php, node.js, CSS та багатьох інших, користувачі пишуть код цих мов між HTML, і він змішується з ними дуже легко.
 6. Щоб програміст був або фронтендером, або бекендером, він повинен мати знання HTML, оскільки це основна мова, а всі інші мови інтегруються з ним. Так само, синтаксис XML - це як HTML та XML, який сьогодні широко використовується для зберігання даних. Якщо хтось добре знає HTML, працювати з XML йому також легко.
 7. Однією з найбільших переваг HTML є те, що можна побачити зміни миттєво, лише збереживши їх та перезавантаживши попередню сторінку HTML. На відміну від інших мов програмування, не потрібно запускати весь код і з'ясовувати, де помилка. Наприклад, якщо ви зробили слово курсивом, воно з'явиться миттєво на сторінці після збереження та перезавантаження.

До мінусів сайтів, які використовують статичні сторінки HTML можна віднести кілька пунктів, які в деяких ситуаціях можуть виявитися досить значними і навіть вирішальними. Ці пункти наведені нижче.

1. HTML може створювати лише статичні та звичайні сторінки, тому якщо користувачам потрібні динамічні сторінки, то HTML не є актуальним.
2. Потрібно написати багато коду для створення простої веб-сторінки.
3. Функції безпеки не дуже надійні в HTML.
4. Якщо користувачам потрібно написати довгий код для створення веб-сторінки, то це створює певну складність.

4.1.4 Java

Java - мова комп'ютерного програмування загального призначення, яка є одночасно клас-орієнтованою, об'єктно-орієнтованою та спеціально розробленою для розробки нового програмного забезпечення для різних платформ. Коли розробник пише програму Java, компільований код (відомий як байт-код) працює в більшості операційних систем (ОС), включаючи Windows, Linux та Mac OS. Java отримала більшу частину свого синтаксису з мов програмування C і C ++. [14].

Java можна знайти де завгодно. Це основна мова для розробки Android, її використовують у веб-додатках, урядових веб-сайтах та технологіях великих даних, таких як Hadoop та Apache Storm. І це також класичний вибір для наукових проєктів, особливо з обробки природних мов. Java була домінуючою для мобільних пристроїв ще в часи перед смартфоном - перші мобільні ігри на початку 2000-х в основному проводилися на Java.

Основний перелік переваг Java наведений нижче.

1. Об'єктно-орієнтоване програмування. Java включає в себе об'єктно-орієнтоване програмування (ООР) – концепцію, що передбачає визначення типу даних, структуру й набір функцій користувачами. Таким чином,

структура даних стає об'єктом, яким можна управляти для створення відносин між різними об'єктами.(рис.4.1)[14]

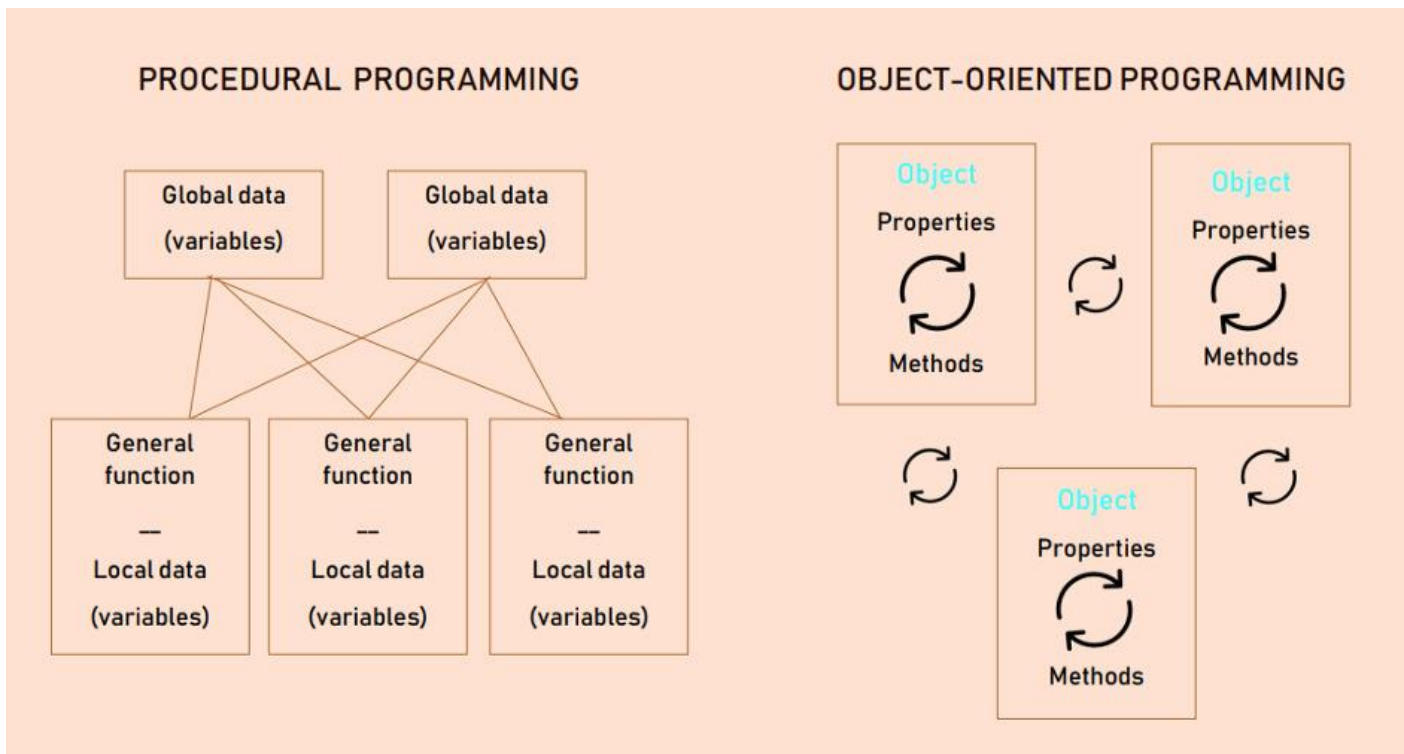


Рисунок 4.1 – Порівняння процедурного та об'єктно-орієнтованого програмування

2. Java є мовою високого рівня, що означає, що вона дуже нагадує людську мову. На відміну від мов низького рівня, що нагадують машинний код, мови високого рівня доводиться конвертувати за допомогою компіляторів чи інтерпретаторів. Це спрощує розробку, полегшує написання, читання та підтримку.
3. Корпоративні програми - це найбільший актив Java. Це почалося ще в 90-х, коли організації почали шукати надійні інструменти програмування, яких не було. Також допомагає широкий набір талантів - Java - мова, яка використовується для впровадження комп'ютерного програмування в більшості шкіл та університетів. Крім того, її інтеграційні можливості вражають, оскільки більшість хостинг-провайдерів підтримують Java. І останнє, але не менш важливе для обслуговування: Java порівняно дешева, оскільки вам не доведеться залежати від конкретної апаратної інфраструктури і ви можете запускати ваші сервери на будь-якому типі машини.

4. Незалежність від платформи («Написати один раз і використовувати скрізь» (WORA)) – популярна в ІТ-сфері фраза, за допомогою якої Sun Microsystems описує багатоплатформні можливості Java. Це означає, що користувач може створити програму Java на, скажімо, Windows, скласти її до байт-коду та запустити додаток на будь-якій іншій платформі, яка підтримує віртуальну машину Java (JVM). У цьому випадку JVM служить рівнем абстракції між кодом і апаратним забезпеченням. Усі основні операційні системи, включаючи Windows, Mac OS та Linux, підтримують JVM. І якщо ви не пишете програму, яка в основному покладається на особливості платформи та інтерфейс користувача, ви можете поділитися - можливо, не всім - але великим шматком байт-коду(рис. 4.2)[14].

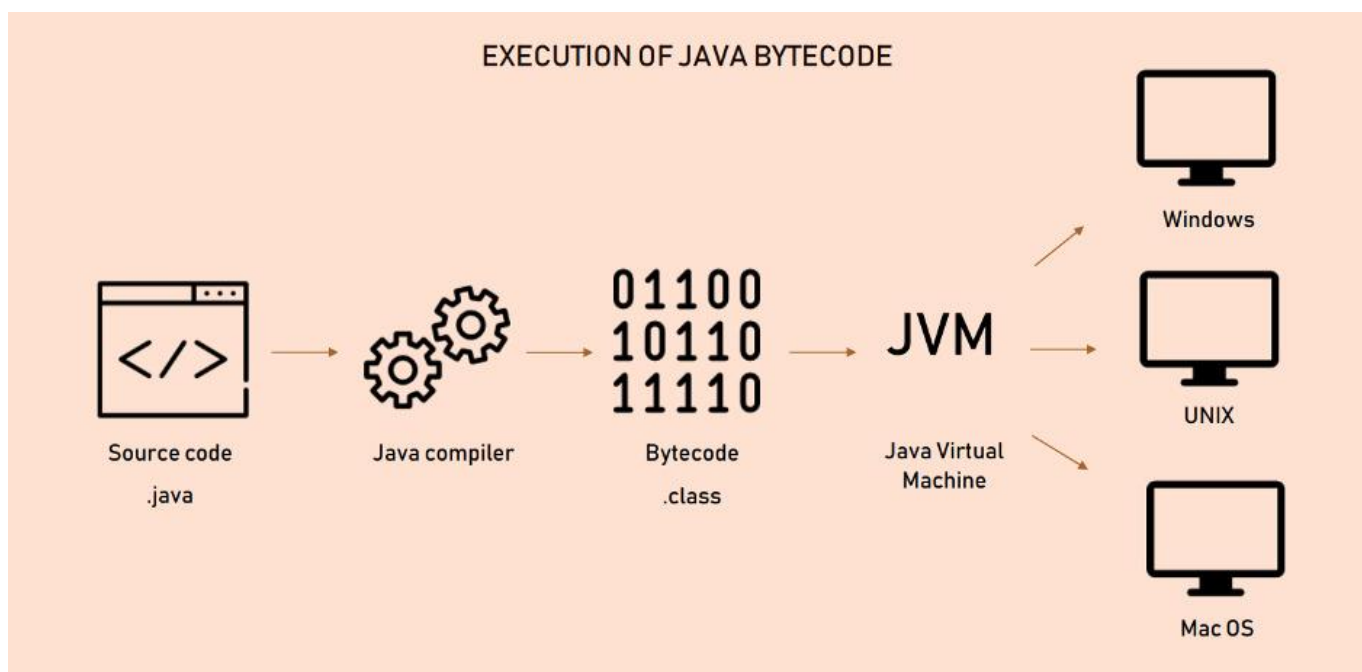


Рисунок 4.2 – Принцип роботи WORA на Java

5. Java була розроблена як розподілена мова, тобто вона має інтегрований механізм обміну даними та програмами між декількома комп'ютерами для підвищення продуктивності та ефективності (рис.4.3)[14].

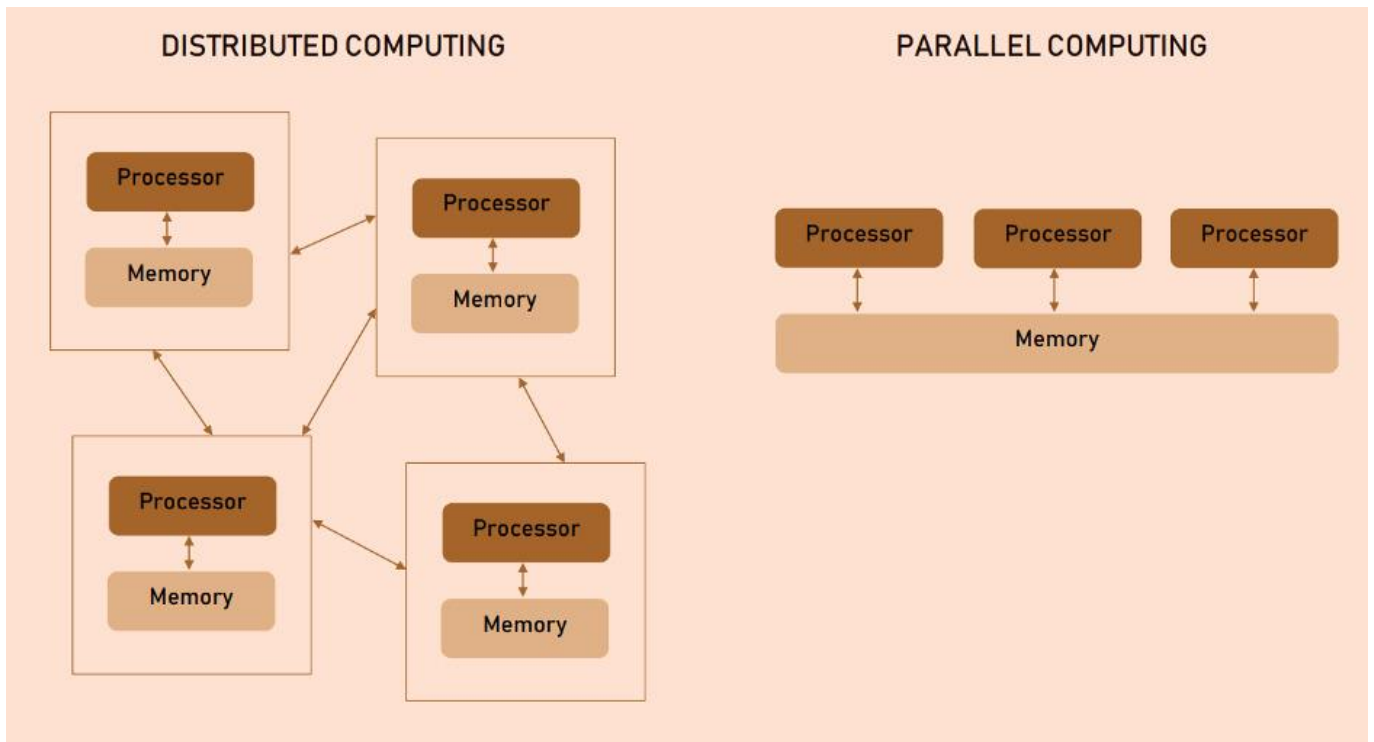


Рисунок 4.3 – Порівняння розподіленого і паралельного програмування

6. **Безпека.** Існує думка, що Java – безпечна мова, але це не зовсім так. Сама мова не захищає користувачів від вразливих місць, проте деякі її функції можуть врятувати від загальних недоліків безпеки. По-перше, порівняно з C, Java не має покажчиків. Вказівник - це об'єкт, який зберігає адресу пам'яті іншого значення, що може викликати несанкціонований доступ до пам'яті. По-друге, в Java є менеджер безпеки, створена політика безпеки для кожної програми, де ви можете вказати правила доступу. Це дозволяє запускати програми Java у "пісочниці", виключаючи ризики заподіяння шкоди.
7. **Автоматичне управління пам'яттю.** Розробникам Java не потрібно турбуватися про те, щоб вручну писати код для завдань управління пам'яттю завдяки автоматичному керуванню пам'яттю (АММ), що також використовується в мові програмування Swift, і збору сміття, додатку, який автоматично обробляє розподіл і розділ пам'яті.
8. **Багатопоточність.** У програмуванні потік - це найменша одиниця обробки. Щоб максимально використовувати час процесору, Java дозволяє запускати ці потоки одночасно - в процесі, який називається багатопотоковим читанням

(рис. 4.4)[14]. Потоки мають однакову область пам'яті, тому перемикання між ними займає мало часу. Вони також є незалежними, тому якщо одна нитка стикається з винятком, вона не впливає на інші потоки. Це особливо корисно для ігор та анімаційних програм.

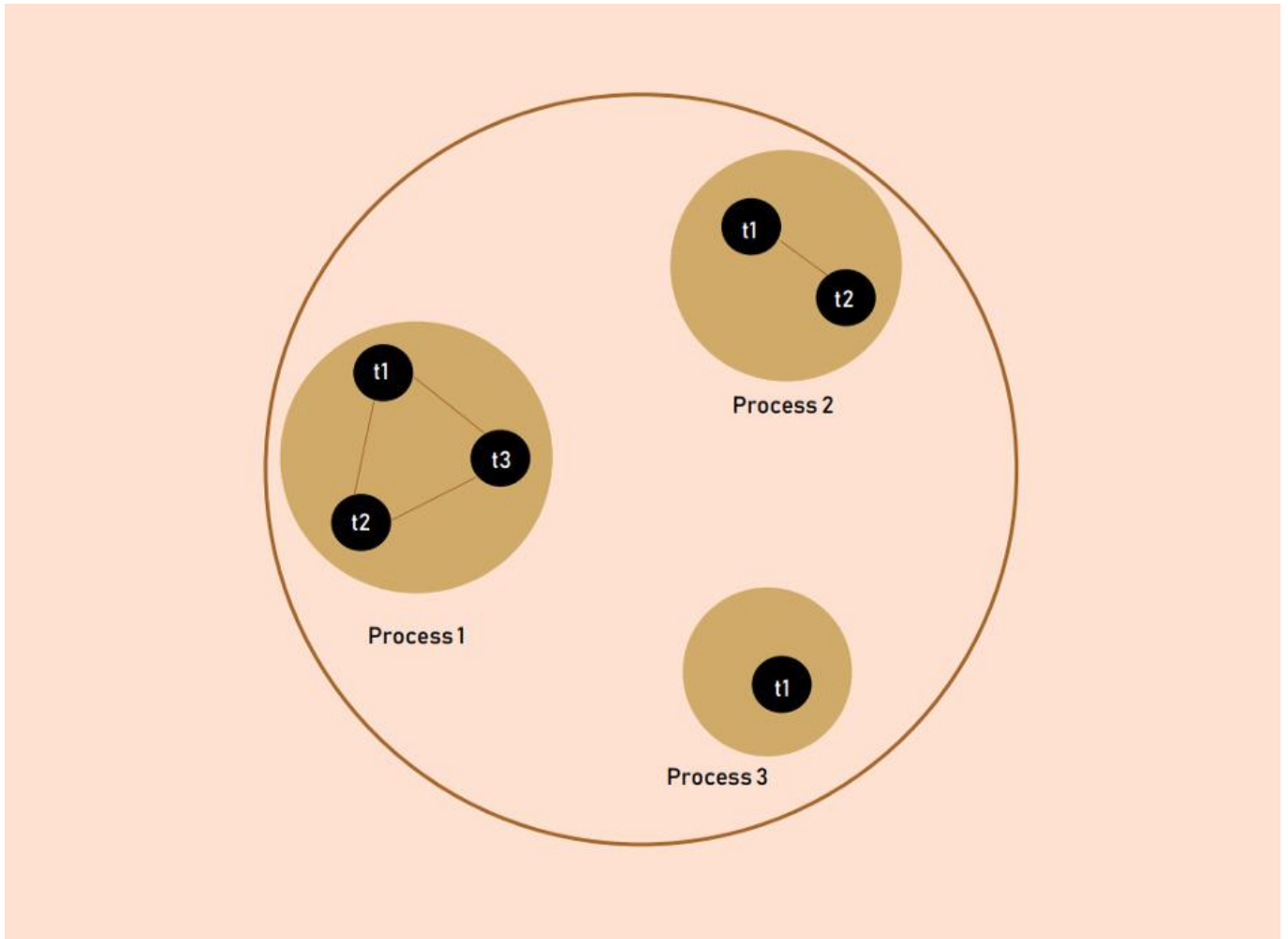


Рисунок 4.4 Приклад багатопоточного виконання

9. Стабільність і співтовариство. Уже багато років розвитку Java сприяють співтовариство, підтримка Oracle і достаток додатків і мов на JVM. Крім того, постійно випускаються нові версії Java з новими цікавими функціями.

Недоліки Java наведені нижче.

1. Нещодавно Oracle оголосила, що вони почнуть стягувати Java Standard Edition 8 для використання в бізнесі, комерції або виробництві починаючи з 2019 року. Щоб отримати всі нові оновлення та виправлення помилок,

користувачам доведеться платити за кількість користувачів або за кількість комп'ютерів.

2. Низька продуктивність. Будь-яка мова високого рівня має справу з низькою продуктивністю через рівень компіляції та абстракції віртуальної машини. Однак це не єдина причина, чому в Java часто критикують швидкість. Наприклад сміттєзбірник - корисна функція, яка, на жаль, може призвести до значних проблем з роботою, якщо це займе більше 20 відсотків часу процесора. Неправильна конфігурація кешування також може спричинити надмірну пам'ять та використання сміття. Існують також тупикові місця, що трапляються, коли кілька потоків намагаються отримати доступ до одного ресурсу, і - кошмар кожного розробника Java - помилки поза пам'яттю. Незважаючи на те, що кожен з цих проблем можна запобігти за допомогою вмілого планування, вони все-таки складаються та можуть призвести до різного обсягу шкоди.
3. Відсутність нативного дизайну. Щоб створити графічний користувацький інтерфейс програми (GUI), розробники використовують різні інструменти, що залежать від мови. Таким чином, для додатків Android існує Android Studio, який допомагає створювати додатки, які виглядають і відчують себе рідними. Однак, що стосується інтерфейсу робочого столу, цього в Java помітно не вистачає.
4. Багатослівний і складний код. Коли код є багатослівним, це означає, що він використовує занадто багато слів. Хоча це може здатися перевагою, коли ви намагаєтесь зрозуміти мову, довгі, занадто складні речення роблять код менш читабельним і зрозумілим. Намагаючись наслідувати англійську, багато мов високого рівня, як правило, містять зайву інформацію. Java, створена для зменшення неприступного C++, змушує програмістів вводити саме те, що вони означають, що робить мову більш прозорою для не експертів, але, на жаль, менш компактною.

4.2 База даних

Для розробки проекту була обрана PostgreSQL. Postgre – Це фактичний двигун бази даних, а SQL - мова програмування, з якою користувач працює для спілкування з базою даних. PostgreSQL - це об'єктно-реляційна база даних, що означає, що зв'язок між двома таблицями підтримується. Будь-яка база даних має формат таблиці, інтегрований з рядками та стовпцями.

PostgreSQL підтримує безліч типів даних, таких як рядок, число, дата та час, як My SQL. Також він підтримує типи даних для геометричних фігур, зображень, мережевих адрес, бітових рядків, пошуку тексту та записів JSON тощо.

Це дуже стара система управління базами даних. Користувач може інтегрувати PostgreSQL з будь-якою мовою програмування, як Java, C, C ++ тощо. Ця функція дозволяє визначити власні спеціалізовані функції. Структурована мова запитів Postgre має багато функцій, які користувачі могли б знайти в інших базах даних.

На наступній схемі (рис.4.5)[15] ми бачимо, що одночасно декілька клієнтів запитують базу даних. І база даних ефективно обслуговує цих клієнтів. Це допомагає PostgreSQL підтримувати цілісність.

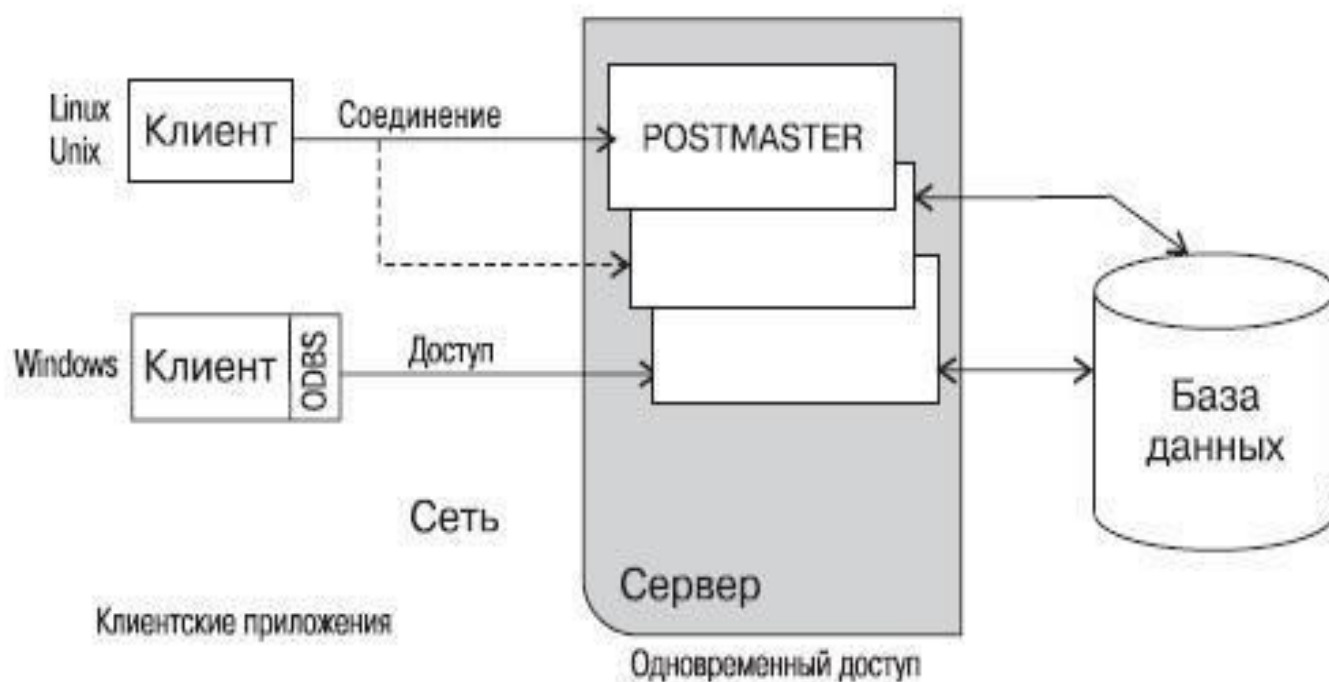


Рисунок 4.5 – Работа типового dodatku PostgreSQL

PostgreSQL має такі переваги:

- простий у використанні;
- має визначений користувачем тип даних;
- відкритий для користувачів;
- велика підтримка громади;
- використовує збережені процедури;
- він підтримує ACID специфіку, тобто атомність, консистенцію, ізоляцію, міцність.

Недоліки PostgreSQL:

- як ми бачимо архітектуру Postgre на наведеній діаграмі(рис.4.5), БД створює окремий сервіс для кожного клієнта, що перетворюється на великий обсяг використання пам'яті;
- якщо робити порівняння, то PostgreSQL не є ліпшим, коли мова йде про продуктивність;
- PostgreSQL не користується великою популярністю, ніж інші системи управління базами даних;
- у PostgreSQL також є брак кваліфікованих фахівців;
- якщо говорити про швидкість, PostgreSQL не гідний суперник порівняно з іншими інструментами.
- зробити реплікацію складніше.
- інсталяція для початківців не проста.

4.3 Фреймворки та бібліотеки

4.3.1 Hibernate

Hibernate - це вільна, відкрита джерельна об'єктно-реляційна бібліотека відображення для Java, призначена для відображення об'єктів у RDBMS та для реалізації об'єктно-орієнтованих концепцій програмування у реляційній базі даних. Цілью Hibernate є звільнення розробника від значного обсягу порівняно

низькорівневого програмування при роботі в об'єктно-орієнтованих засобах в реляційній базі даних. Hibernate в основному вирішує проблеми невідповідності об'єктно-реляційного опору, які виникають, коли реляційна база даних з'єднана програмою, написаною в об'єктно-орієнтованому мові програмування.

Невідповідність об'єктно-реляційного опору виникає через різниці типів даних, маніпулятивні відмінності, транзакційні відмінності та структурні та цілісні відмінності.

У JDBC потрібно написати код, щоб зіставити представлення даних об'єктної моделі на реляційну модель та її відповідну схему. Hibernate відображає класи Java на таблиці баз даних за допомогою XML або за допомогою анотацій.[16]

Hibernate використовує HQL (Hibernate Query Language), схожий на SQL, але HQL Hibernate забезпечує повну підтримку поліморфних запитів. HQL розуміє об'єктно-орієнтовані поняття, такі як успадкування, поліморфізм та асоціація.

Код Hibernate не залежить від бази даних, оскільки користувачу не потрібно змінювати запити HQL (за кількома винятками), коли користувач змінює бази даних, такі як MySQL, Oracle тощо. Отже, легко перейти на нову базу даних. Це досягається використанням дружнього діалекту для спілкування з базою даних. Базу даних можна вказати за допомогою діалекту в конфігурації XML Hibernate. Hibernate зменшує кількість повторних рядків коду, які часто можна зустріти в JDBC. JDBC викине SQLException, що є перевіреним винятком. Таким чином, користувач буде писати блоки "try-catch" у своєму коді. Hibernate обробляє це шляхом перетворення всіх винятків JDBC у неперевірені винятки. Тому вам не потрібно витрачати свій час на реалізацію блоків пробного лову.

З вищезгаданих моментів зрозуміло, що Hibernate обов'язково підвищить продуктивність програми та допоможе скоротити час розробки програми - а отже, і вартість.

4.3.2 Google Maps API

Google Maps API - це надійний інструмент, який можна використовувати для створення власної карти, карти пошуку, функцій реєстрації, відображення синхронізації даних в реальному часі з місцем розташування, планування маршрутів.

Сам сервіс представлений у вигляді набору протоколів, за рахунок яких програмісти і веб-розробники можуть як «по цеглинці» збирати різні додатки. Ще зовсім недавно API був тісно пов'язаний з ОС і додатками на робочому столі. Але за останні кілька років тренд настільки виріс, що API став незамінним інструментом в сфері веб [17].

Сервіс має свої відмінні переваги, які будуть цікаві не тільки програмістам, але і звичайним користувачам. Ничже наведений їх перелік.

1. Простота. Сервіс досить простий і зрозумілий у використанні, користувачеві потрібно всього лише вказати своє місце розташування і прописати дані, які потрібно відобразити на карті. На вказаному місці з'явиться маркер, при натисканні на який відобразиться вся актуальна інформація об'єкта;
2. Хороша візуалізація. Відвідувачі вашого сайту відразу побачать як до вас краще проїхати, або пройти;
3. Доступний функціонал. Сервіс можна оформити під загальну тематику сайту, додатково розширивши його можливості під потреби відвідувачів;
4. Допомогає донести посил компанії до відвідувачів сайту;
5. Підвищує інтерес до продукції і послуг компанії;
6. Клієнти можуть залишати свої відгуки, які будуть відображатися біля контактних даних компанії;
7. Можна додавати актуальну інформацію про всі філії та магазинах компанії, а також про дочірні підприємства.

У підсумку, API Google Maps вважається досить потужною платформою, яка з кожним днем удосконалюється і набирає обертів популярності. Правильне використання всіх функцій сервісу дозволяє зробити будь-яку подорож більш

комфортним, а пошук необхідних адрес і об'єктів стане більш ефективним і швидким.

4.3.3 Vue.js

Vue.js - це прогресивний фреймворк JavaScript, який використовується для створення веб-інтерфейсів та односторінкових додатків. Не тільки для веб-інтерфейсів, Vue.js також використовується як для розробки настільних, так і для мобільних додатків із системою Electron. Розширення HTML та база JS швидко зробили сприятливий інтерфейс інструменту Vue, про що свідчать прийняття такими гігантами, як Adobe, Behance, Alibaba, Gitlab та Xiaomi. [18].

Назва фреймворку - Vue - в англійській мові така ж фонетична, як і view, і відповідає традиційній архітектурі Model-View-Controller (MVC). Простіше кажучи, view - це інтерфейс програми / веб-сайту, а основна бібліотека Vue.js фокусує шар перегляду за замовчуванням. Але MVC не означає, що Vue.js не може використовуватися з іншим архітектурним підходом, як архітектура на основі компонентів (CBA), що використовується в React.

Як і будь-яка технологія, яка набирає популярність, програмування на Vue.js викликає суперечки у всій спільноті. І є причини, чому Vue став другим найулюбленішим фреймворком у 2019 році.

Нижче наведено перелік переваг Vue.js.

1. Крихітний розмір: завантажений zip з фреймворком важить 18 Кб. Як невелика вага, фреймворк не тільки швидко завантажується та встановлюється в бібліотеку, але також позитивно впливає на SEO та UX користувача.
2. Віртуальна візуалізація DOM та продуктивність. Модель об'єкта документа (DOM) - це те, з чим користувач зіткнеться під час надання веб-сторінок. DOM - це представлення HTML-сторінок із його стилями, елементами та вмістом сторінки як об'єктів. Об'єкти, що зберігаються у вигляді структури дерева, генеруються браузером під час завантаження сторінки. Коли

користувач взаємодіє зі сторінкою, об'єкти змінюють свій стан, так що браузер повинен оновлювати інформацію та виводити її на екран. Але оновлення цілого DOM громіздке. З метою швидкості Vue.js використовує віртуальний DOM. Такий підхід робить візуалізацію сторінки досить швидким та покращує продуктивність програми.

3. Реактивне двостороннє прив'язування даних. Ще одна перевага в маніпуляціях з DOM - це двостороння прив'язка даних, успадкована Vue від Angular. Двостороння прив'язка даних - це зв'язок між оновленнями даних моделі та переглядом (UI). Об'єднані компоненти містять дані, які можна час від часу оновлювати. За допомогою двостороннього зв'язування даних простіше оновлювати пов'язані компоненти та відстежувати дані оновлення.
4. Однофайлові компоненти та читабельність. Кожен фрагмент майбутньої програми / веб-сторінки у Vue є компонентом. Компоненти представляють капсульовані елементи інтерфейсу. У Vue.js компоненти можна записувати в HTML, CSS та JavaScript, не поділяючи їх на окремі файли.
5. Інтеграційні можливості та гнучкість. Важливим аспектом будь-якої нової технології є можливість інтеграції з існуючими програмами. З Vue.js це так просто, оскільки він покладається лише на JavaScript і не потребує інших інструментів для роботи.

Недоліки Vue.JS наведені нижче.

1. Мовний бар'єр. Прийняття Vue на таких підприємствах, як Xiaomi та Alibaba, допомогло популяризувати рамки та створило попит на ринку праці. Завдяки популярності Vue.js у Китаї значна частина його змісту та обговорень, не дивно, є китайською.
2. Відсутність підтримки масштабних проєктів. Vue.js - це ще молодий фреймворк. Розмір його спільноти та команди розвитку все ще незрівнянний із більш зрілим Angular. Також не користується фінансовою підтримкою великих підприємств. Щоб бути прийнятою у великих масштабних проєктах,

технологія повинна бути стабільною та сильно підтримуватися, щоб питання можна було швидко вирішити..

Vue.JS реалізує всі сучасні підходи до розробки web UI і є легким в освоєнні, гнучким і високо інтегрованим зі сторонніми технологіями фреймворком. Безумовно, в плані створення складних інтерфейсів (наприклад, для корпоративних сайтів) React його перевершує. Але для рішення щодо простих завдань використання Vue виглядає не тільки виправданим, але і кращим.

4.3.4 Spring Data JPA

Spring Data JPA, що є частиною більшої родини Spring Data, дозволяє легко реалізувати сховища на базі JPA. Цей модуль стосується розширеної підтримки шарів доступу до даних на основі JPA. Це спрощує побудову додатків, що працюють на Spring та які використовують технології доступу до даних.

Реалізація рівня доступу до даних у програмі була громіздкою протягом досить тривалого часу. Для виконання простих запитів, а також виконання сторінок і аудиту потрібно писати занадто багато кодових шаблонів. Spring Data JPA має на меті значно покращити реалізацію шарів доступу до даних за рахунок зменшення зусиль до фактично необхідної кількості. Як розробник користувач пише інтерфейси сховища, включаючи власні методи пошуку, і Spring забезпечить реалізацію автоматично. [19]

Можливості Spring Data JPA:

- витончена підтримка для створення сховищ на основі Spring та JPA;
- підтримка предикатів Querydsl і таким чином безпечних JPA-запитів;
- прозорий аудит доменного класу;
- підтримка сторінки, динамічне виконання запитів, можливість інтегрувати спеціальний код доступу до даних;
- перевірка анотованих запитів @Query під час завантаження;
- підтримка відображення сутності на основі XML;

- конфігурація сховища на основі JavaConfig шляхом введення `@EnableJpaRepositories`.

4.3.5 Spring Security

Spring Security - це потужний і налаштований фреймворк аутентифікації та контролю доступу. Це стандарт фактичного забезпечення додатків на базі Spring.

Spring Security - це структура, орієнтована на надання як аутентифікації, так і авторизації додаткам Java. Як і всі проекти Spring, реальна сила Spring Security виявляється в тому, наскільки легко її можна розширити, щоб відповідати спеціальним вимогам. Ось деякі з можливостей інфраструктури Spring Security:

- декларативна безпека;
- інтеграція API сервлетів;
- комплексна та розширювана підтримка як аутентифікації, так і авторизації;
- підтримка безпеки на рівні методів і анотацій безпеки JSR-250;
- підтримка одноразового пред'явлення пароля;
- підтримка інтеграції контейнерів;
- захист від атак, таких як виправлення сеансу, джек-джей, підробка веб-сайтів тощо.

4.4 Висновок до розділу

У даному розділі було проведено вибір та обґрунтування елементів та технологій, які використовуватимуться для розробки системи, а саме: мови програмування, база даних, фреймворки та бібліотеки, наведено їх переваги та недоліки.

5 СТРУКТУРА БАЗИ ДАНИХ

5.1 ER- діаграма

Метою побудови ER діаграми є створення точного і повного відображення реального світу, що використовується в майбутньому в якості джерела інформації для побудови бази даних. Комплекс завдань цього етапу полягає в виявленні спільних інформаційних об'єктів і зв'язків між ними. Для побудови структури бази даних використовується метод моделювання «Сутність - зв'язок» або ER-діаграма.

При розробці ER діаграми розроблюваної системи було визначено такі сутності, як користувач, група та повідомлення. Один користувач має багато груп, одна група містить багато повідомлень. Приклад ER-діаграми системи представлено на рисунку 5.1 та додатку Д.

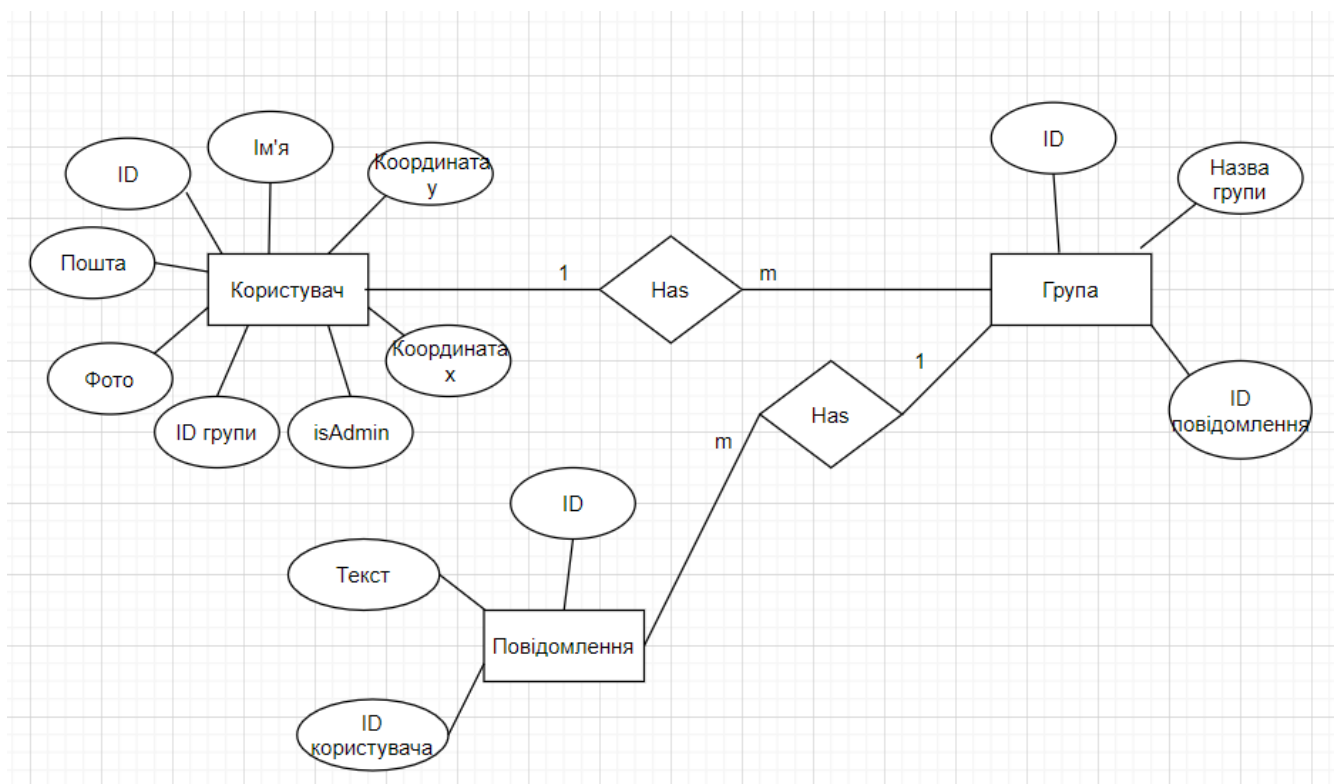


Рисунок 5.1 – ER –діаграма розроблюваної системи

5.2 Схема бази даних

Наступним кроком проектування є створення логічної схеми бази даних, що представлена на рисунку 5.2 та додатку Е .

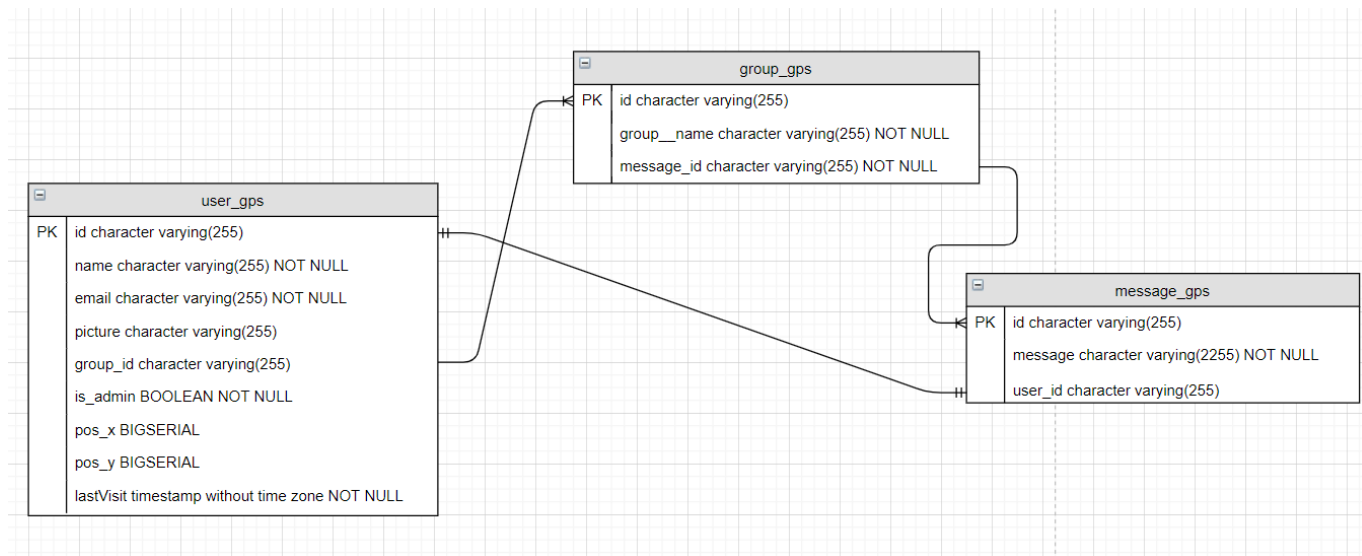


Рисунок 5.2 – Схема бази даних розроблюваної системи

Детальний опис елементів присутній у таблиці 5.1.

Таблиця 5.1

Опис елементів бази даних

Елемент	Опис
Таблиця user_gps	
id	Ідентифікаційний номер користувача, primary key
name	Ім'я користувача
email	Електронна пошта користувача
picture	Фото користувача
group_id	Ідентифікаційний номер групи користувача, foreign key
is_admin	Булеан-змінна, що показує чи є користувач адміністратором
pos_x	Координата X користувача
pos_y	Координата Y користувача
lastVisit	Дата останнього візиту

Продовження таблиці 5.1

Елемент	Опис
Таблиця group_gps	
id	Ідентифікаційний номер групи, primary key
group_name	Назва групи
message_id	Ідентифікаційний номер повідомлення, foreign key
Таблиця message_gps	
id	Ідентифікаційний номер повідомлення, primary key
message	Текст повідомлення
user_id	Ідентифікаційний номер користувача, foreign key

5.3 Висновок до розділу

У розділі розглянута та створена ER -діаграма розроблюваної системи. На основі ER- діаграми була побудована схема бази даних та приведений опис кожного елемента бази даних.

6 АРХІТЕКТУРА ТА ЛОГІКА СИСТЕМИ

6.1 Розробка архітектури системи

Під час розробки системи було вирішено, що вона буде використовувати 3 основні компоненти: веб браузер, веб сервер, базу даних(рис 6.1), AJAX підхід та матиме REST архітектуру.

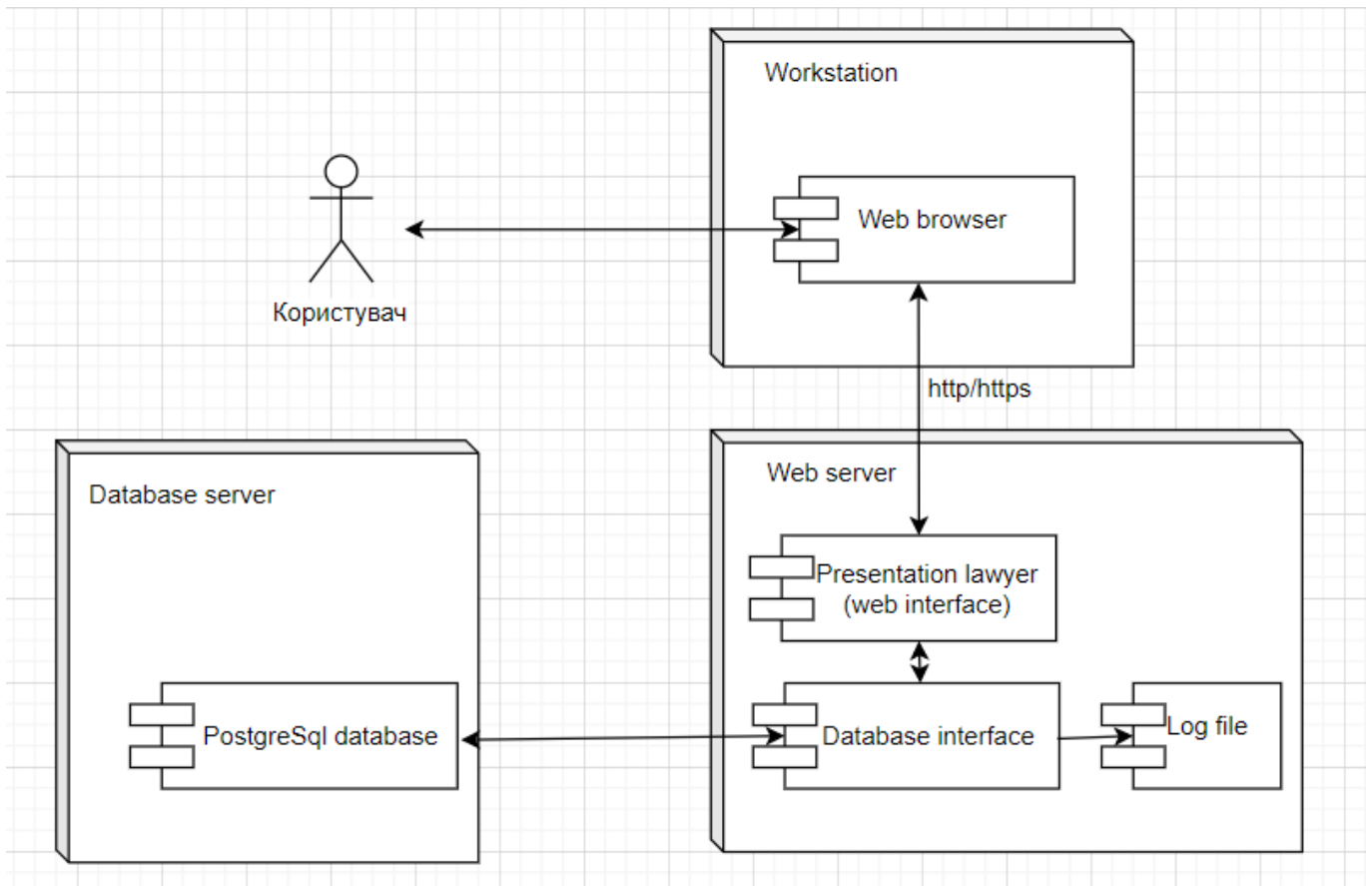


Рисунок 6.1 – Діаграма розгортання

6.1.1 REST

REST – стиль архітектури в мережі для взаємодії компонентів розподіленого додатка. При проектуванні розподіленої системи виникла необхідність у створенні набору обмежень, чим став REST. REST підвищує продуктивність і спрощує архітектуру. Саме через це, цей стиль використовують багато пошукових систем, системи, що працюють з даними та інтернет-магазини В основі, REST використовує

компоненти які взаємодіють як сервери та користувачі у Всесвітній інтернет павутині, що допомагає REST бути альтернативою RPC [21].

Виклик віддаленої процедури може бути звичайним HTTP-запитом в мережі Інтернет (такий запит зазвичай називають «REST-запит»), а дані, які необхідні користувачу передаються у вигляді параметрів запиту.

Для веб-сервісів які використовують REST архітектуру не має офіційного стандарту, які мають сервіси що використовують SOAP. SOAP це протокол, а REST-це архітектурний стиль (рис.6.2)[22].

Сравнительная таблица		
Характеристика	Soap	Rest
Применимы для статических ресурсов	Нет	Да
Применимы для динамических ресурсов	Да	Нет
Сохраняет состояния stateless	Да	Нет
Операции производятся	С помощью методов или операции	Над объектами или ресурсами
Необходим сервер приложений	Да, для разбора XML-запросов и исполнения программного кода	Нет
Затраты на встраивание	Большие	Малые
Масштабируемость	Не большая	Большая
Загрузка вычислительных ресурсов	Большая	Не большая
Изучение дополнительных спецификаций(WSDL, SOAP)	Требуется	Нет, потому что базируется на стандартных Web-протоколах
Стандартизированный механизм доступа к данным	Да, с помощью WSDL	Нет, но разрабатывается WADL – аналог WSDL
Используется шаблон проектирования	Нет	Да – CRUD (4 основных метода GET/PUT/POST/DELETE)
Кэширование на уровне сервера	Нет	Да, с помощью протокола HTTP
Возможность работать с любыми протоколами	Да	Нет, только HTTP
Быстродействие	Нет	Да
Работает с представлением в различных форматах	Нет, только XML	Да
Обработка ошибок	Стандартизирована	Использует известные коды ошибок HTTP

Рисунок 6.2 Порівняння REST та SOAP

REST не являється стандартом, але не дивлячись на це, такі стандарти, як URL, JSON, XML і HTTP використовуються великою кількістю RESTful реалізацій.

Характеристики архітектури REST сторінки:

- масштабованість для забезпечення великої кількості компонентів і взаємодій компонентів;
- продуктивність – з точки зору користувача взаємодія і ефективність компонентів може бути головним фактором продуктивності.

Рой Філдінг - один з головних авторів специфікації протоколу HTTP, описує вплив архітектури REST на масштабованість наступним чином:

- готовність компонентів до можливих змін навіть при працюючій веб-сторінці);
- можливість переносу компонентів системи з переміщенням разом з даними;
- ясність зв'язків між компонентами системи;
- надійність, яка відображається в стійкості до відмови всього сервісу при відмові окремих з'єднань, даних або компонентів;
- зрозумілість інтерфейсу.

Також Філдінг писав, що REST додатками не можуть називати ті додатки, які не мають наведені характеристики. Якщо ж додаток відповідає на всі вимоги і має всі необхідні характеристики, то, він вважає, що додаток буде мати такі переваги:

- масштабованість;
- прозорість системи взаємодії (особливо необхідна для додатків обслуговування мережі);
- надійність (виникає завдяки тому, що зберігати інформацію про стан користувача, яка може бути загублена немає необхідності);
- простота інтерфейсів;
- портативність компонентів;
- продуктивність ;

- легкість внесення змін;
- здатність розвиватись, враховуючи нові вимоги (на прикладі Інтернет мережі).

6.1.2 AJAX

AJAX – термін, який описує нові можливості до побудови інтерфейсів веб-додатків призначених для користувача, При використанні AJAX обмін даними між браузером та сервером відбувається у «фоновому» режимі (рис.6.3)[23].

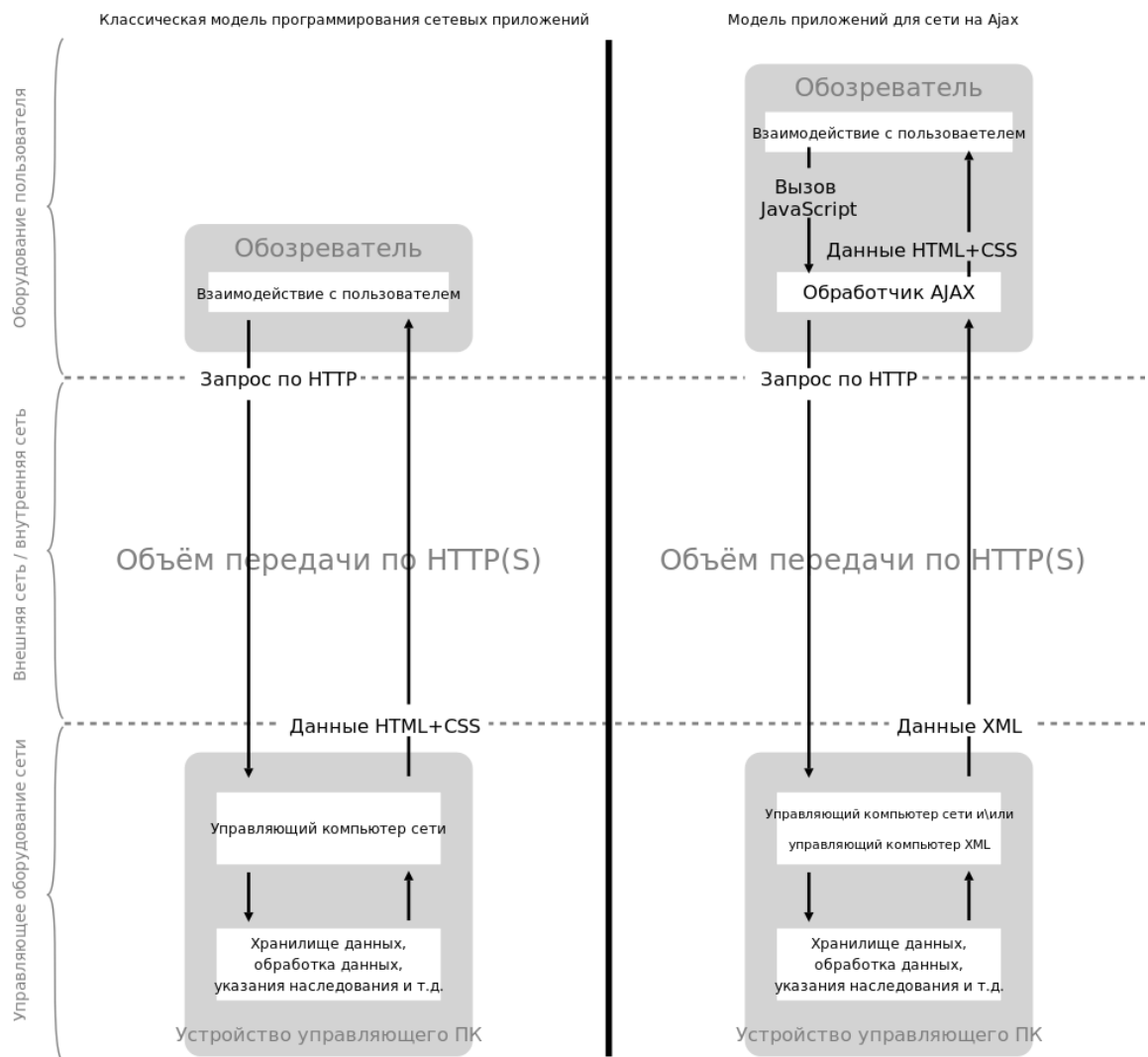


Рисунок 6.3 – Модель класичних додатків для мережі (зліва) в порівнянні із застосуванням AJAX (праворуч).

Веб-додатки працюють швидше та зручніше завдяки тому, що при оновленні даних користувачем веб-сторінка повністю не перезавантажується, а лише ті дані, що були оновлені.

Переваги використання AJAX наведені нижче.

1. Збільшення відповіді інтерфейсу. Користувач може бачити результат своїх дій швидше і без повного перезавантаження сторінки завдяки тому, що завантаження тільки тих даних що змінилась значно швидше.
2. AJAX надає функціонал для інтерактивної обробки. AJAX використовують для розробки адміністративних панелей, чатів та інструментів, які виводять дані, що змінюються з часом. Наприклад, на багатьох веб-сайтах користувачі вводять свої дані при реєстрації і мають можливість одразу бачити чи зайняті ті дані(наприклад логін який може бути унікальним), або при введенні користувачем пошукового запиту і відображенням попередніх запитів та можливих варіантів запитів.
3. Зниження використання трафіку. AJAX дозволяє скоротити використання трафіку завдяки тому, що сторінка яка була оновлена, не перезавантажується повністю, а лише та частина, що була змінена. AJAX також дозволяє змінити дані сторінки після того, як відправить або отримає дані в форматі XML або Json.
4. Оновлення сторінки не зупиняє мультимедіа. AJAX використовує безліч відео- та аудіохостинги тому що при оновленні даних на веб-сторінці відео- та аудіоплеери не зупиняються .
5. AJAX менше завантажує сервер. Коли користувач налаштовує AJAX правильно, це зменшує в кілька разів загрузку серверу. Якщо користувач замінить повне завантаження веб-сторінки генерацією і передачею частини зі змістом, він може забути про такі фактори, що впливають на навантаження на сервер, як довгий час обробки скриптів та запитів до БД, генерація тих сторінок веб-сторінки, які генеруються за одним шаблоном і використовують для цього звернення до різних файлів.

Недоліки використання AJAX наведені нижче.

1. AJAX робить проект складним. AJAX проводить перерозподіл логіки обробки даних, а саме: виділяє і частково переносить на сторону клієнта процеси первинного форматування даних, що робить контроль цілісності форматів і типів складним. Зменшення доступу веб-сторінки для користувачів, збільшення витрат на реалізацію коду та менеджмент проекту може привести до того, що ефект та переваги технології AJAX можуть бути непотрібні та не актуальні.
2. AJAX має повільну швидкість через неправильне використання. AJAX призначений для об'єднання різних технологій та підвищення швидкості їх взаємодії, але при нагромадженні AJAX запитів на одній веб-сторінці, вона починає працювати повільніше звичайної сторінки, що не використовує AJAX.
3. Неможливість сумісного працювання зі стандартними інструментами браузера. Браузер не записує сторінки, що були динамічно створені в історії сторінок, що відвідував користувач, тому кнопка повернення до відвіданих раніше сторінок не працює.
4. Не зручна робота при втраті з'єднання. Існує можливість просто перезавантажити звичайну сторінку при втраті з'єднання. У випадку з AJAX-сторінкою користувачу необхідно перезавантажити сторінку з початку і знайти те місце, де він зупинився. Така особливість AJAX як паралельна робота і так ділить канал на безліч маленьких з'єднань і збільшує ризик розриву каналу.
5. Технології для ведення статистики обліку веб-сторінок стають непотрібними. Багато власників веб-сторінок використовують технології для характеристик та обліку відвідувань своїх нових веб-сторінок. Веб-сторінки з такими технологіями, що використовують AJAX втрачають свою необхідність.
6. В деяких ајах-скриптах існують проблеми з відображенням нестандартного коду.

6.2 Діаграма послідовності

Діаграма послідовності— різновид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.

Для розуміння «спілкування» трьох компонентів системи виникає необхідність побудови діаграм послідовностей основних функцій системи.

6.2.1 Процес авторизації (рис.6.4).

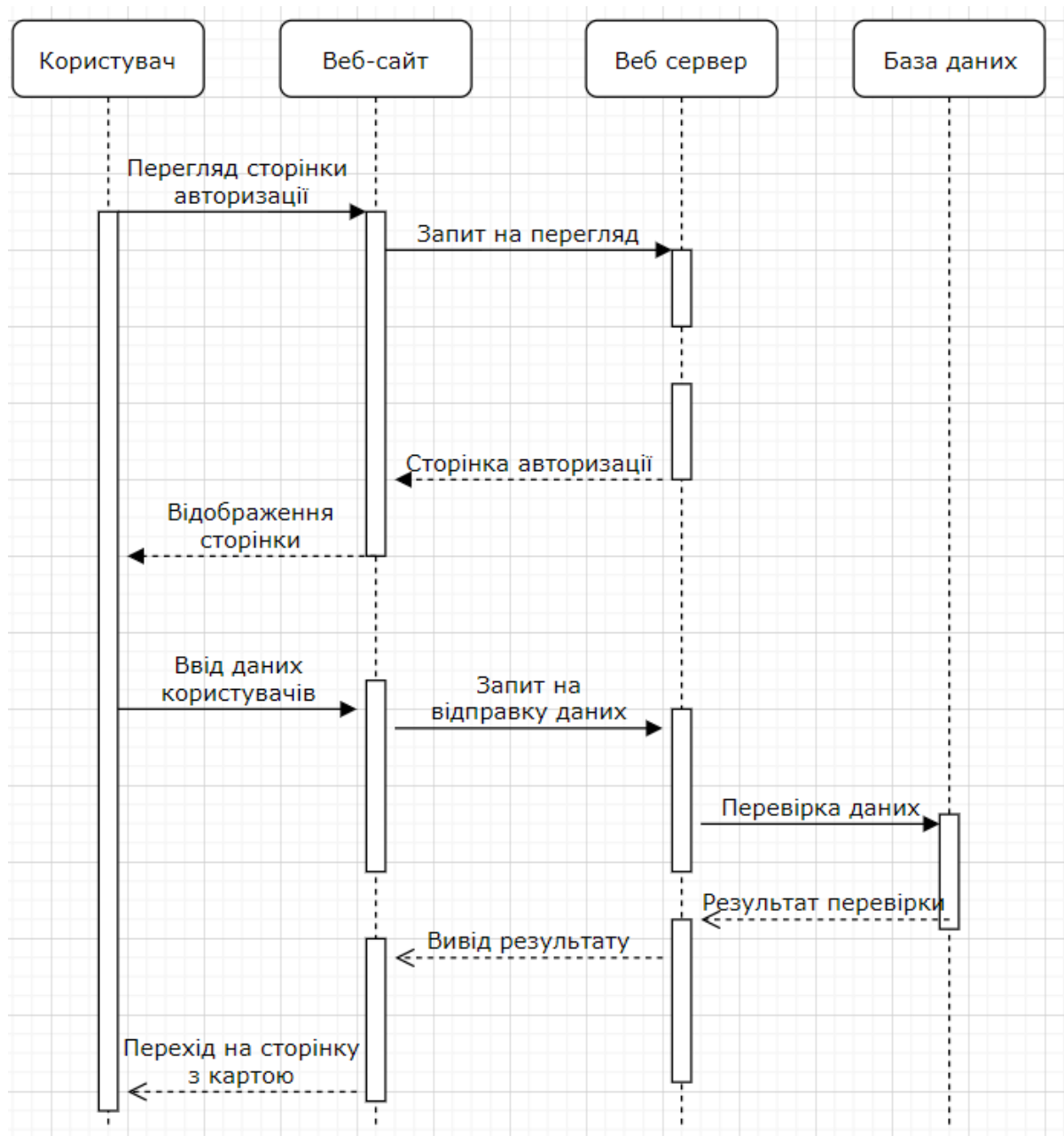


Рисунок 6.4 – Діаграма послідовності процесу авторизації

На діаграмі послідовності процесу авторизації покроково показана послідовність та взаємодія компонентів при авторизації користувача.

6.2.2 Процес реєстрації (рис. 6.5).

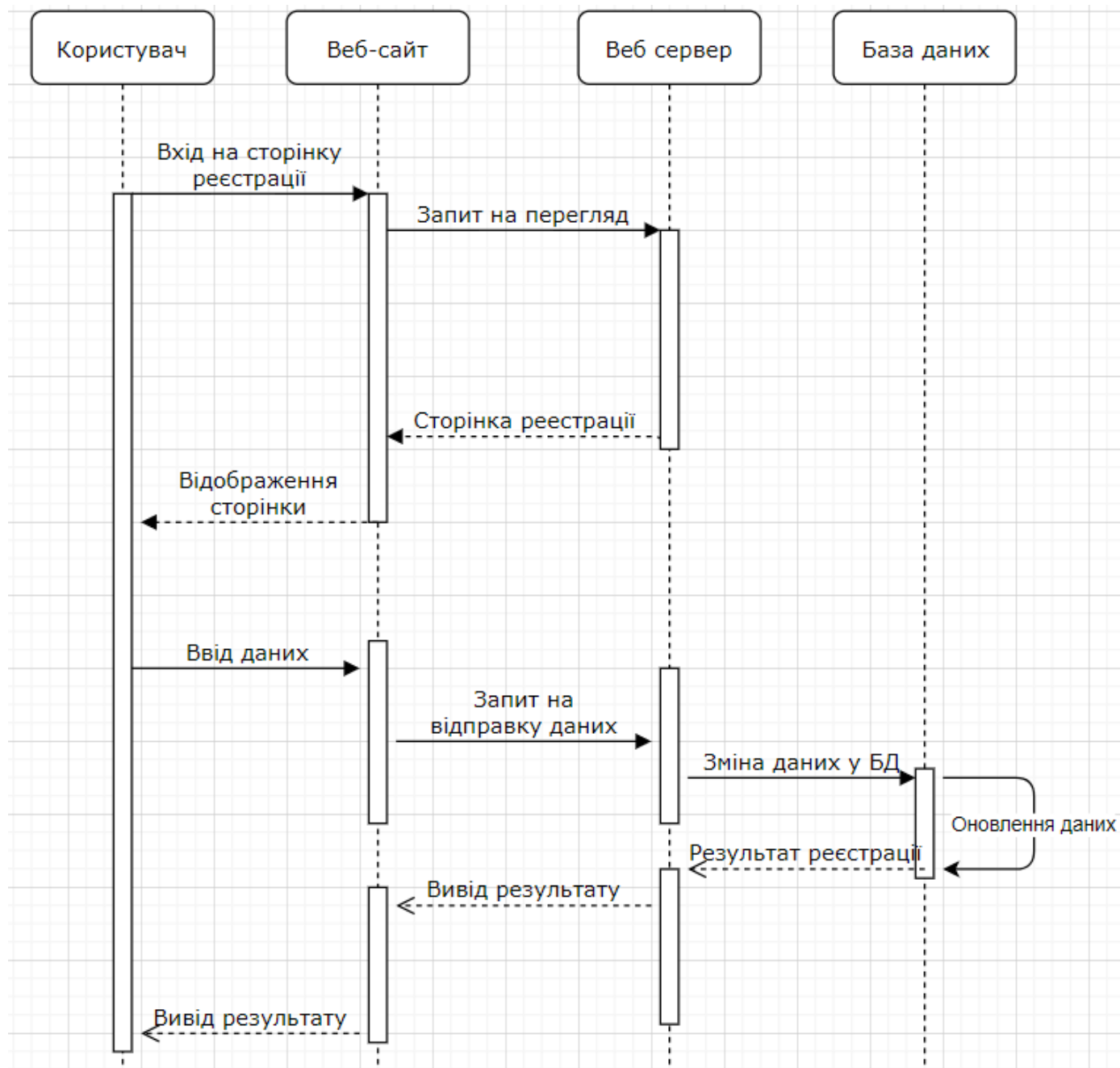


Рисунок 6.5 – Діаграма послідовності процесу реєстрації

На діаграмі послідовності процесу реєстрації покроково показана послідовність та взаємодія компонентів при реєстрації користувача.

6.2.3 Процес визначення місцезнаходження (рис.6.6).

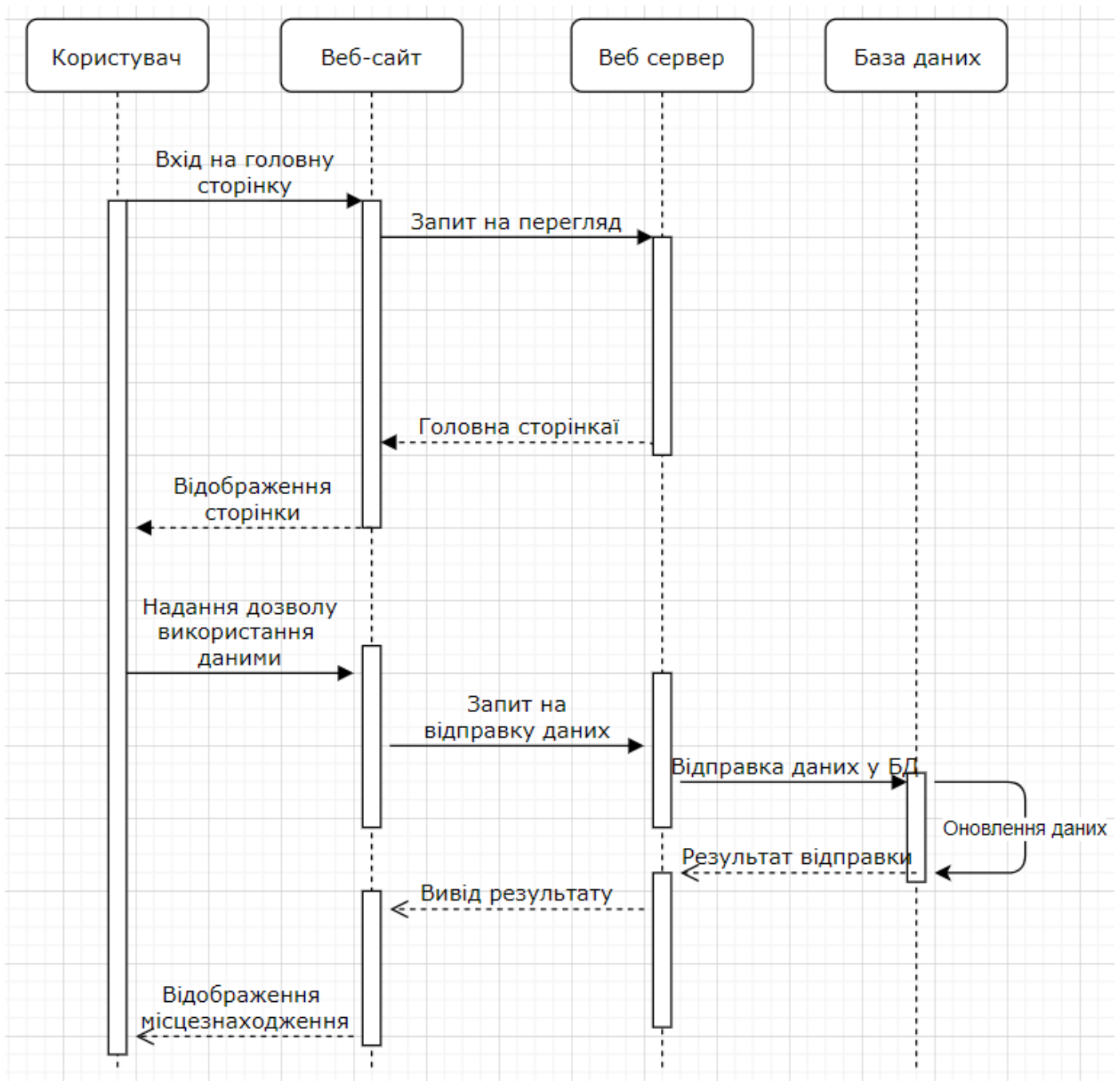


Рисунок 6.6 – Діаграма послідовності процесу визначення місцезнаходження

На діаграмі послідовності процесу визначення місцезнаходження покроково показана послідовність та взаємодія компонентів після входу користувача на головну сторінку.

6.2.4 Процес створення групи (рис.6.7).

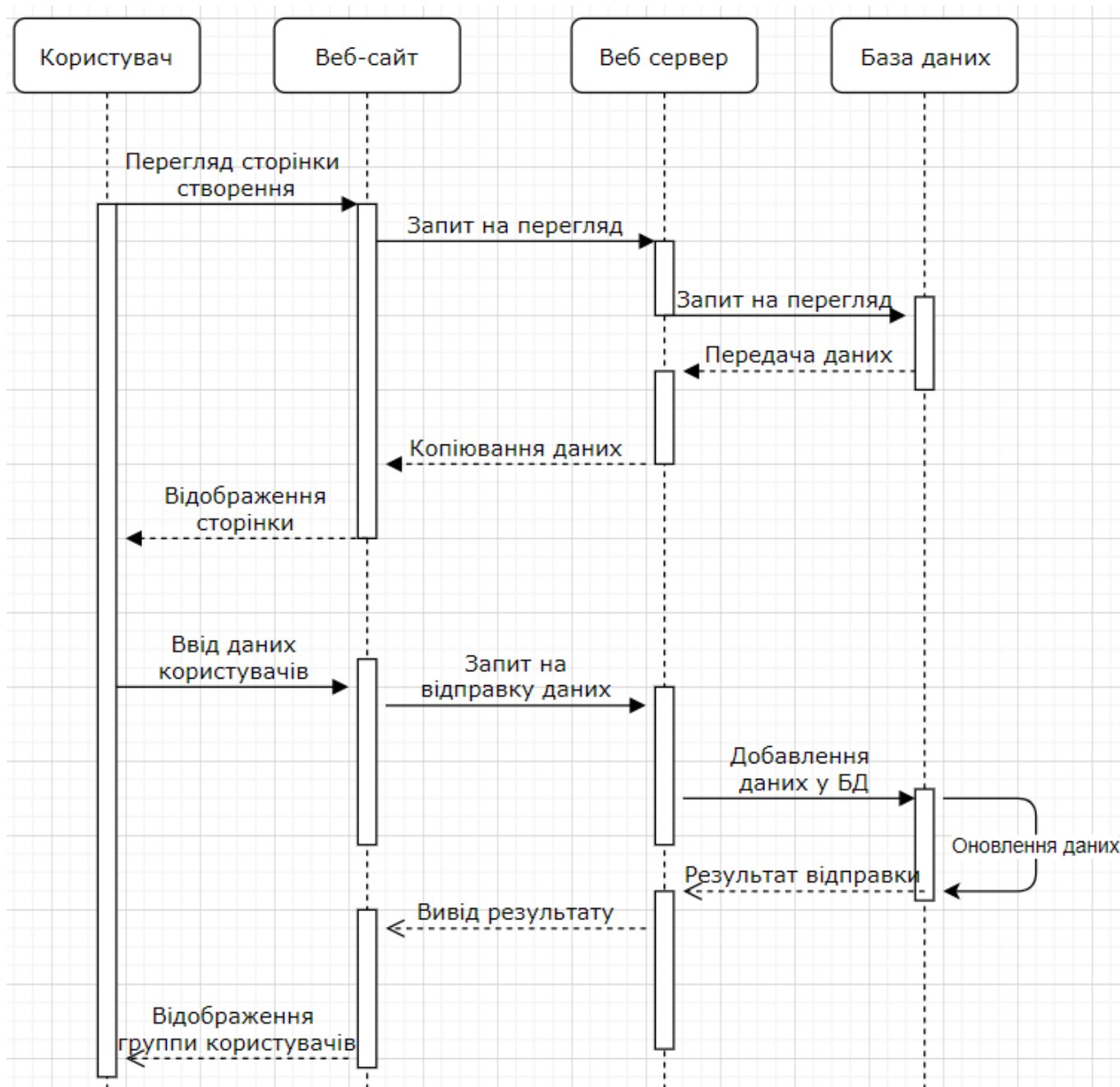


Рисунок 6.7 – Діаграма послідовності процесу створення групи

На діаграмі послідовності процесу створення групи покроково показана послідовність та взаємодія компонентів при створенні користувачем групи.

В ході аналізу діаграм послідовностей стає зрозумілим, що процес створення групи містить більше запитів та є найдовшим. Також після побудови даних діаграм стає зрозуміле «спілкування» трьох основних компонентів системи.

6.3 Шифрування

Для забезпечення збереження даних надалі виникає потреба у шифруванні. Для проекту обрано шифр RSA.

RSA (аббревіатура від прізвищ Rivest, Shamir та Adleman) — криптографічний алгоритм шифрування з відкритим ключем, який на основі використовує складність визначення задачі факторизації великих цілих чисел. RSA визначили як перший алгоритм такого типу, який використовується користувачами як для цифрового підпису, так і для шифрування. Популярні застосунки для шифрування дуже часто базуються на алгоритмі RSA. На рисунку 6.8 [24] показаний приклад шифрування RSA.



Рисунок 6.8 – Приклад роботи RSA шифру

Загалом близько 500 мільйонів користувачів використовують RSA BSAFE бібліотеку шифрування і з кожним разом ця кількість збільшується. Так, як RSA

BSAFE використовує алгоритм шифрування RSA, можна вважати що RSA найбільш розповсюджена криптосистема з відкритим ключем.

В основі алгоритму шифрування RSA лежить програма шифрування PGP, що реалізує хеш-функції(електронний-цифровий підпис). Якщо користувач має відкритий ключ і знає алгоритм шифрування, він не може повторити зашифроване повідомлення. Це є головною перевагою RSA.

6.4 Висновок до розділу

У розділі описано архітектуру системи, метод шифрування, технології, які використовуватиме розроблювана система, їх характеристики, недоліки та переваги. Також у розділі розроблені діаграма компонентів і діаграми послідовностей для різних процесів.

7 СТАРТАП-ПРОЕКТ

7.1 Опис ідеї проекту

В таблиці 7.1 описано зміст ідеї та можливі напрямки застосування.

Таблиця 7.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	1.Страйкбольні ігри,змагання	Користувачі можуть координувати свої дії в режимі реального часу
	2. Військові тренування, бої.	З проведенням необхідних модифікацій та забезпеченням необхідного шифрування систему можуть використовувати військові для координації наземних рухомих об'єктів в тренуваннях та боях
	3.Звичайні користувачі	Користувачі можуть використовувати систему як GPS-навігатор, де вони матимуть змогу ділитись своїми координатами та спілкуватись з ішими користувачами

Продовження таблиці 7.1

	4.Для відслідковування грузового транспорту компаніями	Компанії можуть використовувати систему для відстежування місцезнаходження свого грузового транспорту, кур'єрів.
--	--	---

В ході аналізу потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів була побудована таблиця 7.2.

Таблиця 7.2

Визначення сильних, слабких та нейтральних характеристик
ідеї проекту

№ п/п	Техніко- економічні характерист ики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проект	GPS Лама	MAP-21	Альянс			
1.	Безпека	Шифруван ня	Шифруван- ня	Шифруван ня	Шифруван ня	-	+	-
2.	Координац ія	Чат група	Чат група	Чат група	Відобра- ження на карті	-	-	+
3.	Багатоплат - форменіст ь	Всі платформи	IOS, Android	IOS, Android, Windows	Android, Linux, iOS, Windows,	-	-	+

Продовження таблиці 7.2

4	Інтерфейс	Зрозумілий	Перегружений	Перегружений	Зрозумілий	-	+	-
---	-----------	------------	--------------	--------------	------------	---	---	---

В даній таблиці скорочення W, N, S означають слабку, нейтральну та сильну сторони. Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

7.2 Технологічний аудит ідеї проекту

Для проведення аудиту технології, за допомогою якої можна реалізувати ідею проекту було створено таблицю 7.3.

Таблиця 7.1

Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Реєстрація та авторизація	Spring Security	Технологія наявна, необхідно додати та підлаштувати під проект	Технологія в вільному доступі
2.	Координація	Google Maps Api	Технологія наявна	Технологія у вільному доступі для звичайних користувачів на рік, для продовження та для компаній

Продовження таблиці 7.3

				Необхідно платити
3.	Робота з базою даних	Hibernate	Технологія наявна	Технологія у вільному доступі
		Spring JPA	Технологія наявна, необхідно доробити та підлаштувати під проект	Технологія у вільному доступі
4.	Чат-група	WebSocket	Технологія наявна, необхідно доробити та підлаштувати під проект	Технологія у вільному доступі
5.	Архітектура	Rest	Технологія наявна	Технологія у вільному доступі
		Ajax	Технологія наявна	Технологія у вільному доступі
6.	Інтерфейс	Vue.js	Технологія наявна, необхідно доробити та підлаштувати під проект	Технологія у вільному доступі
Обрана технологія реалізації ідеї проекту: Spring Security, Google Maps Api, Hibernate, Spring JPA, WebSocket, Rest, Ajax, Vue.js.				

7.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку необхідно провести аналіз попиту, що містить в собі наявність попиту, динаміку росту ринку та обсяг. Це вказано в таблиці 7.4.

Таблиця 7.4

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	0
3	Динаміка ринку (якісна оцінка)	Стагнує
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	80%

За результатами таблиці можливо зробити висновок, що ринок є привабливим для входження за попереднім оцінюванням.

В подальшому необхідно визначити потенційні групи користувачів, їх характеристики та сформулювати основний перелік вимог до розроблюваної системи для кожної групи (табл.7.5).

Таблиця 7.5

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп	Вимоги споживачів до товару
1.	Визначення геолокації	Звичайні користувачі.	Звичайні користувачі вирішують свою потребу у визначенні геолокації заходячи на розроблюваний веб-ресурс.	Продукція повинна бути у вільному доступі. Продукція повинна мати не захламлений та зрозумілий інтерфейс для нових користувачів .
2.	Координація за допомогою карти	Звичайні користувачі, компанії, спортивні та страйкбольні організації, військові.	Для всіх цільових аудиторій цієї пореби існують різні фактори, що формують поведінку клієнтів. Звичайні користувачі можуть використовувати продукти безкоштовно, коли всі інші повинні платити.	Продукція повинна відповідати різним вимогам різних цільових аудиторій, бути технічно різною згідно реалізації вимог цільових груп. Продукція повинна мати можливість передачі повідомлень від одних користувачів іншим.

Продовження таблиці 7.5

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп	Вимоги споживачів до товару
3.	Відстеження геолокації різних об'єктів	Компанії та звичайні користувачі.	Компанії повинна платити, звичайні користувачі-ні.	Продукція повинна приймати і правильно оброблювати дані з різних GPS передавачів.

В таблиці 7.6 -7.7 вказані фактори, що сприяють ринковому впровадженню проекту, та фактори, що йому перешкоджають.

Таблиця 7.6

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Нестача управлінського досвіду	Нестача організаційного управління є однією з фундаментальних причин припинення проекту. Через відсутність виробленого алгоритму покрокового	Консультації з експертами у вирішенні необхідних питань

Продовження таблиці 7.6

		вирішення проблем , команда може не переступити навіть ідею посіву	
2.	Фінансування	Помилки зростання в аспектах фінансування присутні головним чином у виборі неправильних стратегій і інструментів фінансування.	Розвиток структури приватного акціонерного капіталу

Таблиця 7.7

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Низька кількість готових рішень	Наразі існує невелика кількість основних конкурентів, що вирішують основні потреби користувачів	Розробка якісного продукту, який міг би затмарити наявні продукти.
2.	Немає обмежень для входу та специфічних вимог до стандартизації	Відсутність обмежень для входу та специфічних вимог до стандартизації сприяє ринковому впровадженню проекту	Розробка продукту з різними та специфічними технологіями

Продовження таблиці 7.7

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3.	Стагнуючий ринок	Застій в сфері координації зумовлює попит на продукти з більшим функціоналом.	Розробити продукт з більшим функціоналом і простим інтерфейсу.

Надалі потрібно провести аналіз пропозиції, де визначаються загальні риси конкуренції(табл.7.8).

Таблиця 7.8

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Чиста конкуренція	Жоден суб'єкт не може впливати на загальну ситуацію на ринку або цей вплив є таким незначним, що ним можна знехтувати.	Розробити якісний продукт,що сподобається користувачам.
2. Глобальний рівень конкурентної боротьби	Продукт може використовуватись по всьому світу	Розробити продукт на різних мовах
3.Внутрішньогалузева конкуренція	Всі продукти діють в межах однієї галузі	Розробити продукт з меншими витратами та

Продовження таблиці 7.8

		ліпшими методами
4. Товарно-родова конкуренція	Розглядається конкуренція з боку товарів-субститутів (замінники). Приклад такої конкуренції - конкуренція між автомобілем і мотоциклом.	Розробити продукт будь-якого виду, але який виконуватиме необхідні функції.
5. Нецінова конкуренція	Боротьба між конкурентами за споживача відбувається шляхом упровадження досягнень науково-технічного прогресу у виробництво, що зумовлює поліпшення якості продукції.	Розробити продукт з ліпшим функціоналом.
6. Не марочна конкуренція	На ринку не існує популярних марок або компаній які вже стали популярними для користувачів	Розробити продукт який буде запам'ятовуватись користувачам

Після аналізу конкуренції необхідно провести більш детальний аналіз умов конкуренції в галузі(табл.7.9).

Таблиця 7.9

Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	«GPS Лама» «Альянс» «МАР 21»	Затрати, ресурси, досвід	Своєчасно випускати оновлення, поліпшувати функціонал	Клієнти будуть обирати кращий продукт з ліпшими характеристиками	Факторів загроз з боку замінників немає
Висновки:	Найсильнішим конкурентом є «GPS Лама»	Потенційних конкурентів немає	Постачальники не диктують умови роботи на ринку	Клієнти не диктують умови роботи на ринку	Обмеження для роботи на ринку через товари-замінники немає

В ході аналізу таблиці можливо зробити висновок, що з огляду на конкурентку ситуацію продукт що розробляється може легко увійти на ринок та буде мати сильні сторони характеристики, щоб бути конкурентоспроможним.

На основі аналізу попередніх таблиць необхідно визначити та обґрунтувати перелік факторів конкурентоспроможності (табл.7.10).

Таблиця 7.10

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1.	Багатоплатформеність	Багатоплатформеність впливає на кількість користувачів, які можуть використовувати продукт. Якщо продукт призначений лише для одної платформи-це знижує кількість користувачів.
2.	Зрозумілість інтерфейсу	Чим легше користувачеві використовувати і розуміти інтерфейс продукту, тим більше шанс, що він буде використовувати продукт на постійній основі та порадить його своїм знайомим.
3.	Безпека	Захист персональних даних є невід’ємною частиною продуктів, що використовують такі дані. Продукти з надійною безпекою більше цінуються користувачами.
4.	Доступність для нових користувачів	Доступність для нових користувачів підрозуміває те, що звичайний користувач, який вперше зайшов на розроблюваний продукт – міг би відразу розібратись з модулями системи, з тим, як правильно користуватись системою. Доступність – це час, за який новий користувач отримує доступ до сервісу.

Після визначення факторів конкурентоспроможності необхідно провести аналіз сильних та слабких сторін проекту(табл. 7.11).

Таблиця 7.11

Порівняльний аналіз сильних та слабких сторін системи для координації наземних рухомих об'єктів за допомогою GPS

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з системою для координації наземних рухомих об'єктів за допомогою GPS						
			-3	-2	-1	0	+1	+2	+3
1.	Багатоплатформеність	19	+						
2.	Зрозумілість інтерфейсу	16				+			
3.	Безпека	15					+		
4.	Доступність для нових користувачів	18			+				

Для детального аналізу можливостей впровадження проекту необхідно скласти SWOT – аналіз (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities)(табл. 7.12).

Таблиця 7.12

SWOT – аналіз стратап-проекту

Сильні сторони: Багатоплатформенність, Доступність для нових користувачів, зрозумілість інтерфейсу	Слабкі сторони: Безпека
Можливості: Розробити якісний продукт з ліпшими характеристиками ніж існуючі аналоги, який буде користуватись популярністю через стагнуючий ринок	Загрози: Можлива розробка більш якісного продукту конкуруючою коомпанією

На основі SWOT – аналізу необхідно розробити альтернативи ринкової поведінки для виведення стартап-проекту на ринок(табл. 7.13).

Таблиця 7.13

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Самостійна розробка	80%	3-6 місяців
2.	Виготовлення додаткового програмного забезпечення для існуючих	60%	4-8 місяців
3.	Розробка на вимогу компанії-замовника	30%	3-6 місяців
4.	Розробка для спеціальних спортивних організацій	60%	7-10 місяців

Після аналізу альтернатив ринкового впровадження було обрано самостійну розробку, так як ймовірність отримання ресурсів та строки реалізації є найбільш вигідними.

7.4 Розроблення ринкової стратегії проекту

Для початку розроблення ринкової стратегії необхідно визначити стратегію охоплення ринку, а саме: опис цільових груп потенційних споживачів (табл. 7.14). Після проведення аналізу потенційних груп споживачів необхідно обрати цільові групи та визначити стратегію охоплення ринку.

Таблиця 7.14

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Приватні компанії	Готові	Високий попит	Досить велика конкуренція	Входити у сегмент просто
2.	Звичайні користувачі	Готові	Високий попит	Середня конкуренція	Входити у сегмент просто
3.	Спортивні та страйкбольні організації	Готові	Високий попит	Низька конкуренція	Входити у сегмент просто
4.	Військові	Готові	Високий попит	Висока конкуренція	Входити у сегмент дуже складно
Які цільові групи обрано: Приватні компанії, звичайні користувачі, спортивні та страйкбольні організації.					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл.7.15).

Таблиця 7.15

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Створення програмного продукту	Диференційован ий маркетинг	Розроблюваний проект буде містити більший та кращий функціонал	Стратегія диференціації

Надалі необхідно обрати стратегію конкурентної поведінки(табл. 7.16).

Таблиця 7.16

Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
Ні	Компанія буде шукати нових споживачів та	Так, а саме: визначення місцезнаходження	Стратегія лідера

Продовження таблиці 7.16

	забирати існуючих у конкурентів		
--	------------------------------------	--	--

На основі даних з попередніх таблиць необхідно розробити стратегію позиціонування (табл. 7.17) , що полягає у формуванні ринкової позиції, за якими споживачі мають ідентифікувати проект.

Таблиця 7.17

Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспромо жні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Зручне визначення місцезнаходження, можливість координації між користувачами, визначення координат наземних рухомих об'єктів	Стратегія диференц іації	Продукт з кращим та розширеним функціоналом	Безкоштовно для звичайних користувачів, низька ціна на підписку для компаній Зручне та швидке вирішення проблеми Використання продукту споживачами на постійній основі

7.5 Розроблення маркетингової програми стартап-проекту

Для початку необхідно сформувати маркетингову концепцію товару який отримає споживач. Для цього необхідно підсумувати результати аналізу конкурентоспроможності товару (табл. 7.18).

Таблиця 7.18

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Визначення місцезнаходження	Товар вирішує потребу споживачів надаючи необхідні дані у зручному вигляді	Безкоштовно для звичайних користувачів, зручний інтерфейс та простота вирішення потреби
2.	Координація груп користувачів за допомогою карти	Товар пропонує вирішення потреби споживачів та надає зручні методи для координації	Безкоштовно для звичайних користувачів, зручний інтерфейс та простота координації груп користувачів
3.	Відстеження місцезнаходження різних об'єктів	Товар надає можливість користувачам відстежувати координати об'єктів	Необхідно реалізувати таку перевагу над конкурентами, як можливість товару працювати в «режимі сну» та передавати своє місцезнаходження серверу

В подальшому необхідно розробити трирівневу маркетингову модель товару в якій уточнюється ідея продукту та його послуги, його фізичні складові, особливості процесу (табл. 7.19).

Таблиця 7.19

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Координація наземних рухомих об'єктів за допомогою GPS		
	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Веб-ресурс	Нм	Тл
	2. Наявність шифрування		
	Якість: Стала робота програиноного забезпечення		
	Пакування: відсутнє		
	Марка: кафедра АУТС КПІ ім. Сікорського		
	До продажу: Документація		
	Після продажу: підтримка та оновлення продукту		
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок захищеного доступу до кодової бази, та захисту інтелектуальних прав.			

Надалі необхідно визначити цінові межі якими необхідно керуватись при встановленні ціни на потенційний товар для таких цільових груп як компанії (табл. 7.20).

Таблиця 7.20

Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
-	0-1000	Високий	0-1000

В таблиці 7.21 показано визначення оптимальної системи збуту.

Таблиця 7.21

Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнти можуть купити ліцензію або оформити підписку	Оновлення товару, обслуговування клієнтів, надання документації	Канал нульового рівня	Розповсюдження через Інтернет

В таблиці 7.22 представлена концепція маркетингових комунікацій.

Таблиця 7.22

Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонуванн я	Завдання рекламного повідомлення	Концепція рекламного звернення
Придбання товару або оформлення підписки для тривалого використання	Інтернет, прямі, офіційні	Зручність Надійність Багатоплатфо рменість	Розповісти про переваги розроблювано го продукту	Презентація демо-відео з перевагами товару

7.6 Висновок до розділу

У розділі описано розробку стартап проекту та отримано результати щодо доцільності розробки системи координації наземних рухомих об'єктів за допомогою GPS. Також, у розділі визначено та обґрунтовано такі результати:

- наявна можливість ринкової комерціалізації проекту, присутній попит, динаміка та рентабельність роботи на ринку;
- визначена висока конкурентоспроможність проекту, низький бар'єр входження та наявна перспектива впровадження з огляду на потенційні групи клієнтів;
- подальша імплементація проекту є доцільною;
- обрана альтернатива впровадження.

ВИСНОВКИ

Під час виконання магістерської дисертації було розроблено програмне забезпечення системи координації наземних рухомих об'єктів з допомогою GPS.

У першому розділі було представлено предметне середовище, проведений пошук і аналіз наявних аналогів, в ході якого було визначено характеристику та функціонал розроблюваної системи.

У другому розділі були сформовані та описані функціональні та не функціональні вимоги, описаний процес діяльності, сформовані цілі й призначення розробки. Також було побудовано структурну схему та описані бізнес правила.

У третьому розділі описано й проаналізовано основний сценарій використання системи та розглянуто всі прецеденти.

У четвертому розділі присутній обґрунтований вибір елементів та технологій що були обрані для реалізації розроблюваної системи, а саме: мови програмування, база даних, фреймворки та бібліотеки, наведено їх переваги та недоліки .

У розділі структури бази даних описано та побудовано ER- діаграму, а на основі неї побудовано схему бази даних.

У розділі архітектури та бізнес-логіки системи описано розробку архітектури системи, створено діаграми послідовностей, обґрунтовано вибір технологій архітектури системи та обрано методу шифрування.

В розділі стартап-проекту описана ідея проекту, проведений технологічний аудит ідеї проекту та аналіз ринкових можливостей запуску стартап-проекту. Також в розділі присутня розробка ринкової стратегії проекту та маркетингових програм.

Магістерська дисертація відповідає вимогам завдання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Сайт з інформацією про геолокацію [Електронний ресурс]/ Режим доступу: [https://ru.wikipedia.org/wiki/ Геолокация_\(информатика\)](https://ru.wikipedia.org/wiki/Геолокация_(информатика))
2. Сайт з інформацією про тріангуляцію [Електронний ресурс]/ Режим доступу: <https://mobile-review.com/articles/2009/triangulation.shtml>
3. Сайт з інформацією про тваринні трекери [Електронний ресурс]/ Режим доступу: <http://selliall.fun/gps-treker-dlya-sobak>
4. Сайт з інформацією про застосунок «Где Я» [Електронний ресурс]/ Режим доступу: <https://gdeya.net/>
5. Сайт з інформацією про застосунок «Enduro Tracker» [Електронний ресурс]/ Режим доступу: <https://enduro-tracker.ru.aptoide.com/>
6. Сайт з інформацією про застосунок «GPS Лама» [Електронний ресурс]/ Режим доступу: <https://play.google.com/store/apps/details?id=com.levstone.mobility.grouptracker&hl=ru>
7. Сайт з інформацією про СН-3210 «Базальт-К» [Електронний ресурс]/ Режим доступу: <https://orizon-navigation.ua/ru/product/aparatura-nazemnogo-priznachennya/>
8. Сайт з інформацією про систему «Андромеда-Д» [Електронний ресурс]/ Режим доступу: <https://dfnc.ru/c106-technika/andromeda-d-nadezhnaya-sistema-upravleniya-vojskami/>
9. Сайт з інформацією про систему «Альянс» [Електронний ресурс]/ Режим доступу: <http://yourtactic.com/news/view/19>
10. Сайт з інформацією про бізнес правила [Електронний ресурс]/ Режим доступу: http://consulting.ru/econs_art_163768820
11. Сайт з інформацією про JavaScript [Електронний ресурс]/ Режим доступу: <https://www.ipipe.ru/info/javascript.html>
12. Сайт з інформацією про CSS [Електронний ресурс]/ Режим доступу: https://www.prostobiz.ua/biznes/biznes_start/stati/css_dlya_nachinayuschih_istoriya_poyavleniya_plyusy_i_minusy

13. Сайт з інформацією про HTML [Електронний ресурс]/ Режим доступу:
<https://www.rserdyuk.ru/html-vs-cms.html>
14. Сайт з інформацією про Java[Електронний ресурс]/ Режим доступу:
<https://medium.com/nuances-of-programming/плюсы-и-минусы-программирования-на-java-2861f4c2a0d5>
15. Сайт з інформацією про PostgreSQL[Електронний ресурс]/ Режим доступу:
<https://oracle-patches.com/common/3214-%D1%87%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-postgresql>
16. Сайт з інформацією про Hibernate[Електронний ресурс]/ Режим доступу:
<http://javastudy.ru/interview/jee-hibernate-questions-answers/>
17. Сайт з інформацією про Google Maps Api[Електронний ресурс]/ Режим доступу:
<https://maps-creator.com/blog/what-is-google-maps-api.html>
18. Сайт з інформацією про Vue.js[Електронний ресурс]/ Режим доступу:
<https://ru.wikipedia.org/wiki/Vue.js>
19. Сайт з інформацією про Spring Data Jpa[Електронний ресурс]/ Режим доступу:
<https://spring.io/projects/spring-data-jpa>
20. Сайт з інформацією про Spring Security[Електронний ресурс]/ Режим доступу:
https://ru.wikibooks.org/wiki/Spring_Security/
21. Сайт з інформацією про REST[Електронний ресурс]/ Режим доступу:
<https://ru.wikipedia.org/wiki/REST>
22. Сайт з інформацією про REST[Електронний ресурс]/ Режим доступу:
<https://html-templates.info/blog/znakomtes-arhitektura-REST>
23. Сайт з інформацією про AJAX[Електронний ресурс]/ Режим доступу:
<https://ru.wikipedia.org/wiki/AJAX>
24. Сайт з інформацією про RSA[Електронний ресурс]/ Режим доступу:
<https://uk.wikipedia.org/wiki/RSA>